Thesis for The Degree of Doctor of Philosophy

Collaborative, Intelligent, and Adaptive Systems for the Low-Power Internet of Things

Valentin Poirot

Division of Computer and Network Systems
Department of Computer Science & Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2022

Collaborative, Intelligent, and Adaptive Systems for the Low-Power Internet of Things
Valentin Poirot

ISBN 978-91-7905-686-5
Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie nr 5152
ISSN 0346-718X
Technical Report 223D

Department of Computer Science & Engineering
Division of Computer and Network Systems
Network and Systems Unit
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2022.

"There is no meaning in life a priori. Life is nothing until it is lived; but it is yours to give it a meaning, and the value of it is nothing but the meaning that you choose."

– Jean-Paul Sartre

Collaborative, Intelligent, and Adaptive Systems for the Low-Power Internet of Things

Valentin Poirot
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

With the emergence of the Internet of Things (IoT), more and more devices are getting equipped with communication capabilities, often via wireless radios. Their deployments pave the way for new and mission-critical applications: cars will communicate with nearby vehicles to coordinate at intersections; industrial wireless closed-loop systems will improve operational safety in factories; while swarms of drones will coordinate to plan collision-free trajectories. To achieve these goals, IoT devices will need to communicate, coordinate, and collaborate over the wireless medium. However, these envisioned applications necessitate new characteristics that current solutions and protocols cannot fulfill: IoT devices require consistency guarantees from their communication and demand for adaptive behavior in complex and dynamic environments.

In this thesis, we design, implement, and evaluate systems and mechanisms to enable safe coordination and adaptivity for the smallest IoT devices. To ensure consistent coordination, we bring fault-tolerant consensus to low-power wireless communication and introduce Wireless Paxos, a flavor of the Paxos algorithm specifically tailored to low-power IoT. We then present STARC, a wireless coordination mechanism for intersection management combining commit semantics with synchronous transmissions. To enable adaptivity in the wireless networking stack, we introduce Dimmer and eAFH. Dimmer combines Reinforcement Learning and Multi-Armed Bandits to adapt its communication parameters and counteract the adverse effects of wireless interference at runtime while optimizing energy consumption in normal conditions. eAFH provides dynamic channel management in Bluetooth Low Energy by excluding and dynamically re-including channels in scenarios with mobility. Finally, we demonstrate with BlueSeer that a device can classify its environment, i.e., recognize whether it is located in a home, office, street, or transport, solely from received Bluetooth Low Energy signals fed into an embedded machine learning model. BlueSeer therefore increases the intelligence of the smallest IoT devices, allowing them to adapt their behaviors to their current surroundings.

Keywords

Internet of Things, IoT, Low-Power Wireless Networks, Synchronous Transmissions, Consensus, Bluetooth Low Energy, Adaptive Networking, TinyML

Acknowledgment

A journey that started five years ago and spanned two different countries has now come to its natural conclusion. I'm grateful to have met so many great colleagues and made such extraordinary friends along the way.

First of all, I would like to thank my advisor, Olaf Landsiedel, for allowing me to take on this journey. Your mentoring helped me grow both scientifically and personally over the years. I really enjoyed our discussions, collaboration, and your inexhaustible yet invaluable feedback when it comes to writing.

I would also like to extend my thanks to the colleagues I met and friends I've made along the way. At Chalmers, I would like to thank Christos for his philosophical discussions, Dimitris for constantly singing, even more so when prompted to stop, Georgia for her never-ending good humor, Romaric for representing France abroad, as well as Bastian, Karl, Magnus, Fazeleh, Thomas R., Francisco, Tomas, Elad, Marina, Philippas, Vincenzo, Thomas P., Amir, and Hannaneh. Special thanks to Beshr and Babis for being great members of the team. Of course, working and studying at Chalmers would not have been possible without the precious help of Monica, Rebecca, Clara, Eva, and Marianne.

In Kiel, I would like to thank Oliver, with whom I started the PhD journey in Sweden and traveled across the globe, Patrick, for being an outstanding office mate, Janek, for the great discussions and ice-cream breaks, Steffi, Gerd, and Brigitte, as well as the new generation: Birkan, Marc-Andre, Tayyaba, and Ali. In France, special thanks to Arnaud for the gaming sessions, Paul-Lou and Julien for sharing the PhD experience from afar, and scattered across the globe, the PERCCOM family.

Finally, I cannot finish before thanking the people that helped me grow into who I am today, my lovely parents and my sister, who saw me getting closer at every step since the frozen Swedish Lapland, but never quite within reach, for their unconditional support along every step I've chosen. *Merci à tous !*

Valentin Poirot
Kiel, August 2022

List of Publications

Appended publications

This thesis contains the following publications:

[A] **V. Poirot**, B. Al Nahas, O. Landsiedel
Paxos Made Wireless: Consensus in the Air
Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN), 2019, pp. 1–12.
This paper was nominated as a *candidate to the best paper award.*

[B] P. Rathje, **V. Poirot**, O. Landsiedel
STARC: Low-power Decentralized Coordination Primitive for Vehicular Ad-hoc Networks
Third International Workshop on Intelligent Transportation and Connected Vehicles Technologies (ITCVT), part of: IEEE/IFIP Network Operations and Management Symposium (NOMS), 2020, pp. 1–6.

[C] **V. Poirot**, O. Landsiedel
Dimmer: Self-Adaptive Network-Wide Flooding with Reinforcement Learning
Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), 2021, pp. 293–303.

[D] **V. Poirot**, O. Landsiedel
eAFH: Informed Exploration for Adaptive Frequency Hopping in Bluetooth Low Energy
Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 2022, pp. 1 – 8

[E] **V. Poirot**, O. Harms, H. Martens, O. Landsiedel
BlueSeer: AI-Driven Environment Detection via BLE Scans
Proceedings of the ACM/IEEE Design Automation Conference (DAC), 2022, pp. 871–876.
This paper was nominated as a *candidate to the best paper award.*

Other publications

The following publications were published during my PhD studies, or are currently in submission/under revision. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not related to the thesis.

[a] **V. Poirot**, M. Ericson, M. Nordberg, K. Andersson
"Energy efficient multi-connectivity algorithms for ultra-dense 5G networks"
in Springer Wireless Networks, 2020, (26) pp. 2207—2222.

[b] **V. Poirot**, O. Landsiedel
"Poster: Learning to Shine - Optimizing Glossy at Runtime with Reinforcement Learning"
Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN), 2019, pp. 226-227.

Contents

List of Figures

List of Tables

Part I

Introduction

1

Introduction

The emergence of the Internet of Things (IoT) brings communication capabilities to classical embedded systems. Whether via wireless radios or wired networking, these networked systems, previously only composed of memory, computation, and input/output peripherals, can now exchange information and cooperatively solve complex tasks. For example, cyber-physical systems in factories rely on IoT devices to monitor and control industrial processes in real-time [1], while swarms of drones cooperate to fly in formations and avoid collisions [2]. These connected devices form the basis of the Internet of Things, a computing paradigm where objects and machines are augmented with computing power and communicate locally or over the Internet.

Today, we continue to see a strong growth in the adoption and deployment of wireless-enabled IoT systems: four billion new Bluetooth devices were shipped in 2020 alone [3], network operators served nearly two billion active cellular IoT connections in 2021 [4], and we expect 200 million connectivity-augmented vehicles to be driving around in 2025 [5]. Two current trends can explain the rise of IoT deployments across the globe: first, the wide availability and low cost of embedded components, which saw a sharp increase in computing capabilities over the years while maintaining their low power consumption; and secondly, an increasing demand for data, driven by the need for more efficiency, measurability, and transparency.

The prospects of communication-able devices deployed everywhere are leading to the birth of new and more complex distributed applications: we expect cars to form platoons with nearby vehicles, coordinate at intersections or while merging incoming traffic, and to become partly or fully autonomous in the future. Smartwatches equipped with heart rate sensors and smart glucose-meters prelude the development of full on-body sensor networks that will help practitioners monitor patients from afar and alert them of emergencies in real-time. Resource-limited devices such as IoT cameras will be expected to execute deep machine learning models and will require processing support from neighboring systems and edge devices. Local interaction thus becomes paramount for wireless IoT systems: devices locally need to communicate, coordinate, and collaborate to solve complex tasks.

These new functionalities will be expected from all types of IoT devices: from cars with large computing resources down to drones and deployed actuators relying on small batteries. In this thesis, we refer to the smallest and least-powerful IoT devices as *Low-Power IoT*. These devices are marked by severe limitations on all fronts:

- Limited energy budget, usually by relying on small batteries, with an expected operational lifetime measured in tens of hours (earphones) up to months (sensors in remote areas);
- Limited computing resources, currently $\leq$ 5 MB RAM and $\leq$ 100 MHz CPU and in general, several order of magnitudes below typical computers;
- Limited bandwidth, with datarate usually below 5 Mbit/s; and
- Unreliable communication.

Collaborative IoT. In parallel to the growth of IoT, cloud computing has taken an important place in today's computing landscape. Many IoT deployments send all the data they produce to the cloud for storage and processing. However, constantly relying on the cloud for high-performance computation is not a luxury all low-power IoT systems can afford: the low datarates and packet losses induce high communication delays that are impractical in time-critical applications. For instance, drone coordination cannot constantly probe the cloud to compute quick evasive maneuvers. Instead, low-power IoT devices must rely on their embedded resources and local interaction to achieve their goals. Coordination over unreliable communication thus becomes a prime objective for future low-power IoT systems.

Adaptive IoT. Yet, the continual growth of deployed IoT devices leads to an overcrowding of the shared wireless spectrum. Each new device deployed brings an additional traffic burden and must share the wireless resources with past, present, and future deployments. Along with an increased risk of message collisions over the air, the superposition of all local traffic results in an unpredictable environment to external observers: the wireless medium acts inherently as a dynamic environment. The presence of mobile devices, whether in the form of humans using smartphones while walking or mobile machines, further reinforces the dynamicity of tomorrow's wireless environment. New IoT systems must integrate mechanisms to detect, react, and adapt to changes to their wireless environments, both within their network stack and at the application level.

1.1 Motivation and Goals

Wireless IoT systems provide support for a wide range of applications, from headphones and city-wide sensor deployments up to tomorrow's transportation system and Industry 4.0. Within the low-power IoT paradigm, many of those applications must, however, operate with limited energy supplies and should maximize their operational lifetime between two battery charges. In addition to energy constraints, low-power wireless systems must also deal with unreliable communication. For the past decades, the scientific and industrial communities

designed communication standards [6, 7], methods [8], and protocols [9] to improve reliability over lossy links and multi-hop deployments. Some of those techniques, such as synchronous transmissions, achieve up to 99.99% reliability over multiple hops while remaining energy-efficient. However, most protocols remain best-effort in nature and can suffer from degraded performance under sudden and unaccounted external interference.

Challenges. Distributed applications building upon coordination between members, such as swarms of drones maneuvering or industrial closed-loop control, require strict guarantees from their coordination mechanisms: participants must agree on a unique decision, as inconsistencies can have drastic impacts on the operational safety, e.g., in industrial environments. In low-power IoT, the coordination or agreement process shall support communication failures, such as packet losses and delayed receptions, and participant failures, such as devices crashing due to depleted energy storage or simply unable to maintain communication due to mobility. While fault-tolerant consensus is a well-studied field in wired networks, such as in datacenters, most proposed solutions are often deemed unfit to work over low-power wireless communication due to high communication and overhead costs. The first challenge tackled by this thesis therefore relates to providing network-wide agreement with guarantees in the context of low-power IoT, where communication and devices are subject to failures.

A second challenge stems from user mobility and the wireless medium's dynamic and unpredictable behavior. As the number of IoT deployments grows, so does the use of wireless resources. External data traffic, especially aperiodic or bursty, causes the medium to act as a dynamic environment, where communication performance cannot be accurately predicted and can quickly degrade as external interference arises. This is particularly problematic in low-power wireless communication, where transmissions are often over-powered by concurrent WiFi signals and where transmissions are kept to a minimum to save energy [10]. Further, some IoT devices subject to mobility challenges see their environment constantly changing as they move around, and should adjust their behavior accordingly. For example, phones should automatically turn silent mode on once entering a theater, while wireless headsets should limit noise cancellation near roads to ensure the safety of their users. To tackle this challenge, we argue for adaptive IoT systems equipped with mechanisms able to recognize their environment, detect changes within, and able to autonomously react to maintain performance even under unforeseen perturbations. Such mechanisms should be implemented as part of the wireless network stack, for example, by updating protocol parameters at runtime, or can act as a middleware informing applications of sudden changes.

Goals. From these starting observations, our goals are two-fold:

1. We want to enable complex coordination for low-power IoT systems. Therefore, we tackle the problem of network-wide fault-tolerant consensus, where both communication and participants can fail. Specifically, the coordination mechanism must provide consistency guarantees even in the presence of such failures.

2. We aim to improve the adaptivity and resilience of wireless systems. Therefore, we must give IoT devices the ability to detect changes to their

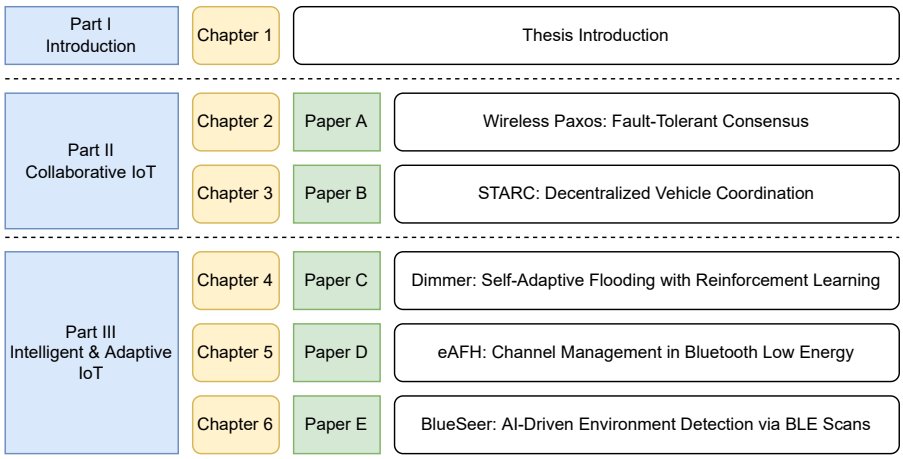

Figure 1.1. Structure of the thesis. Part I introduces the motivation and goal of the work, Part II covers the challenge of collaborative IoT, and Part III tackles the challenge of intelligent and adaptive IoT.

wireless environments and enable them with mechanisms to react to them, for example, by updating protocol behaviors in their wireless stack. We do so both via traditional methods and by relying on recent advances in machine learning, therefore bringing intelligence to low-power IoT and wireless networking.

Approach. In this thesis, we use experimental computer science methods. We design, implement, and evaluate systems, typically network protocols or middlewares, to bring this vision of a collaborative, intelligent, and adaptive IoT to a state of reality. We make our design and source code freely accessible to enable their use and improvement by the community.

Outline. This thesis is a collection of five articles published over the past years and is organized into three parts. Part I provides an introduction to the topic covered in this thesis. It starts with a motivation and overview of the field, background information necessary to understand the appended publications, and a survey of the current state-of-the-art. It then provides a research statement, lists the scientific contributions compiled in this thesis, and concludes the overview of this work. Part II includes two articles and investigates the problem of consensus and coordination in low-power IoT. Part III compiles three articles and covers the topic of intelligent and adaptive IoT. Fig. 1.1 depicts how this thesis is structured.

1.2 Background

This section introduces the core concepts this thesis builds upon. First, we provide an overview of low-power wireless communication, with a focus on short-range technologies such as IEEE 802.15.4 and Bluetooth Low Energy, as well as a primer on synchronous transmissions. Secondly, we present standard

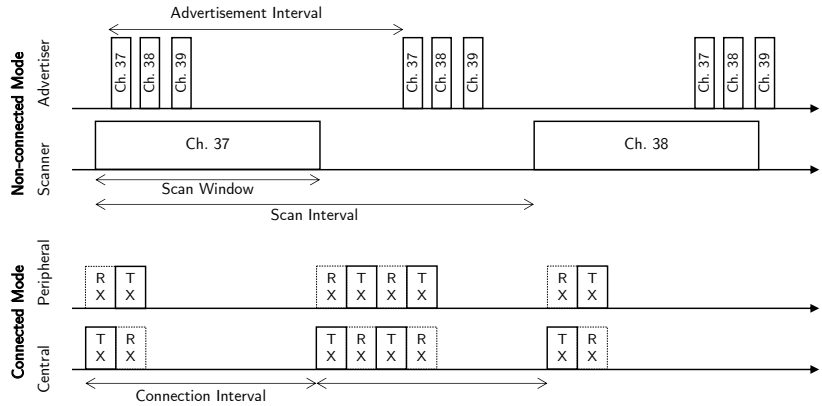

Figure 1.2. Communication in Bluetooth Low Energy. In non-connected mode, an advertiser pseudo-periodically advertises its presence on all advertisement channels while a scanner listens for packets to connect to. In connected mode, the central and peripheral communicate periodically, hopping between frequencies for each connection event.

problem definitions and algorithms for consensus in distributed systems. Finally, we provide a brief introduction to embedded intelligence, with a focus on tiny machine learning and reinforcement learning.

1.2.1 Low-Power Wireless Communication

Due to their limited energy supplies, low-power wireless IoT devices often rely on energy-efficient radios and low-power standards restricting communication to either low datarates or limited coverage. Low-power wide area networks (LPWAN), such as LoRa [11] and Sigfox [12], aim to connect battery-powered devices over large areas; they provide ranges up to a few kilometers but severely limit datarates down to tens of kbit/s [11]. In contrast, wireless personal area networks (WPAN) provide short-range communication, usually tens of meters, with datarates up to a few Mbit/s. IEEE 802.15.4 [6] and Bluetooth Low Energy (BLE) [7] are the two most prominent standards for narrowband short-range networking in use today. Both operate on the 2.4 GHz ISM band, but IEEE 802.15.4 also supports sub-GHz bands and provides ultra-wideband support above 3 GHz.

IEEE 802.15.4. IEEE 802.15.4 is a widespread standard specifying the physical and medium access control layers for low-rate WPAN. Zigbee, WirelessHART, and Thread are common network protocols built upon IEEE 802.15.4. The standard targets reliable wireless communication for home automation and industrial environments. In the 2.4 GHz band, the standard uses an O-QPSK modulation scheme, Direct Spread Spectrum Sequence (DSSS) for forward error correction, and provides a datarate of 250 kbit/s. It divides the ISM band into 16 2-MHz channels and specifies two medium access mechanisms: CSMA/CA and time-slotted channel hopping (TSCH). Devices usually form mesh or tree architectures to communicate over multiple hops.

Table 1.1. Conditions for successful synchronous transmissions in 802.15.4-2.4 GHz.

Effect	Constructive Interf.	Capture Effect
Identical Data?	Identical	Possibly different
Power difference	-	≥ 3 dB
Time difference	≤ 0.5 μs	≤ 160 μs

Bluetooth Low Energy. Introduced as part of Bluetooth 4.0 in 2010 and now in version 5.3 [7], Bluetooth Low Energy (BLE) mainly targets point-to-point communication and local broadcasting. Although it shares its name with Bluetooth Classic, it is a radically different protocol. BLE uses a GFSK modulation scheme and provides datarates ranging from 125 kbit/s in coded modes up to 2 Mbit/s in uncoded modes. It divides the spectrum into 40 2-MHz wide frequency channels, 37 reserved for data traffic and 3 for advertisements and broadcasts. BLE has two operation modes: connected and non-connected mode. The non-connected mode is used to locally broadcast data, disclose the presence of connectable devices, or initialize a connection between a central device (e.g., smartphone) and a peripheral (e.g., headphones). Once a connection is established, the devices periodically communicate on the 37 dedicated data channels via frequency hopping, as depicted in Fig. 1.2. Bluetooth Mesh, a third protocol introduced by the Bluetooth Special Interest Group, builds upon BLE to provide mesh communication by relying on BLE advertisements combined with a flooding mechanism to disseminate messages over multiple hops.

Synchronous Transmissions. Wireless radios are broadcast-oriented innately: any antenna in the vicinity of a transceiver receives a radio signal if the channel conditions are favorable. When two transmissions overlap, their radio waves add up, often leading to an illegible signal at the receptive end. We refer to this physical behavior as *destructive* interference. For long, all overlapping transmissions were considered destructive and avoided at all costs. However, when two transmitters transmit *identical data* at the *same time*, the resulting identical radio waves superpose in what is called *constructive* interference and can be successfully decoded by the receiver. Ringwald and Römer showed the first use of such constructive superposition, a concept named Synchronous Transmissions (ST), with a protocol called BitMAC, by superposing On-Off Keying symbols [13]. Later, A-MAC extended the concept from bits to small packets by concurrently acknowledging request messages [14]. Glossy goes a step beyond and provides network-wide, fast floods supporting mobile nodes without the need for expensive routing [9]. To work in IEEE 802.15.4, constructive interference requires a synchronization error smaller than 0.5 μs, which corresponds to an IEEE 802.15.4 chip period.

Later works softened the claims of constructive interference. Wilhehm et al., as well as Liao et al., argue that what we observe is rather non-destructive interference: the received signal can be decoded but is nonetheless degraded by the collision [15,16]. The DSSS coding scheme used in the physical layer of IEEE 802.15.4 is hypothesized as the reason packets survive such interference. Yet, Al Nahas et al. empirically show that BLE-based synchronous transmissions are

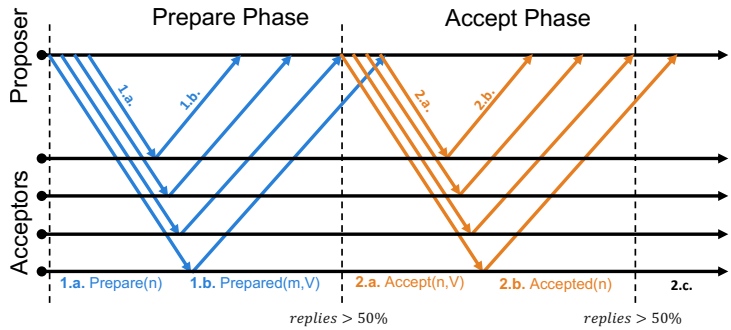

Figure 1.3. Solving consensus with Paxos. A proposer sends a Prepare request with proposal number n. Acceptors reply with the last accepted proposal m with value V. After a majority of Prepared(), the proposer adopts V and sends an Accept request. Proposal n is then accepted by the acceptors.

feasible with Blueflood [17], a BLE-based Glossy counterpart, both for coded and uncoded BLE transmission [17, 18]. Lobba et al. also demonstrate the feasibility of synchronous transmissions using IEEE 802.15.4 ultra-wideband radios [19].

Capture Effect. While synchronous transmissions provide an efficient and fast method to build one-to-all dissemination mechanisms in low-power wireless networking, the data-equality requirements of constructive interference can hinder the design space of higher-level protocols building upon the concept. Instead, we can rely on a second physical behavior to loosen the ST requirements: the capture effect. In IEEE 802.15.4, a signal can be distinguished from concurrent transmissions and successfully decoded if its received-signal strength is at least +3 dB higher than the sum of all concurrent signals. The capture effect supports scenarios where each transmitted data is different but requires that the strongest signal is received no later than 160 μs after a receiver radio first picks up a concurrent transmission, which corresponds to the on-air preamble duration. Chaos builds upon the capture effect and in-network processing to deliver efficient network-wide data aggregation [20], while Mixer integrates network coding to achieve efficient concurrent many-to-many data sharing [21]. With uncoded BLE transmissions, a successful capture effect requires a signal strength difference of +7 dB, which complicates the design of capture-based protocols [17].

1.2.2 Distributed Consensus Algorithms

In the field of distributed computing, a distributed system is composed of participants distributed across a network that collaborate to execute a common task. The ability to reach a decision, defined as a *consensus*, is a fundamental problem of distributed systems [22]. To reach a consensus, a group of participants must reach an agreement on a single data item. Coordination of drones, distributed databases, leader election, and state-machine replication are few examples of applications where consensus is required. In fact, multiple problems in distributed systems can be reformulated as consensus problems,

and thus solved if we solve consensus [23]. A correct solution to the consensus problem must have the following properties:

- *Validity:* the agreed value has been initially proposed;
- *Agreement:* all correct processes agree on the same, unique value;
- *Termination:* every process decides in a bounded time; and
- *Integrity:* if all correct processes choose value v, then any correct process must choose v.

FLP impossibility. It is common in distributed systems to specify the assumptions used on the processes and network by relying on well-defined models. In a synchronous system, all processes execute in lock-steps, or rounds. Within a round, a process can receive messages, execute a computation step, and send messages. Processes have, therefore, a known upper bound on their step time and message delivery is also bounded. In contrast, the asynchronous model assumes no upper time bound on process computation and message delivery.

One of the most influential results in distributed systems is known as the FLP impossibility [22]. Fischer et al. prove that in an asynchronous system with crash failures, it is impossible to design an algorithm for consensus fulfilling all the above properties. As message deliveries are finite but unbounded, it is impossible to distinguish between a failed process and a delayed message. This leads to possibly infinite executions, and validity, agreement, and termination cannot be all satisfied together. To solve consensus, we must therefore adopt at most a partially synchronous model: although the system behaves asynchronously at first, eventually, the system will have bounded process step-time and message delivery time.

Commit and process-failures. A common sub-category of consensus is known as the commit problem, and is often found in distributed databases. In the commit problem, a group of participants must either agree to all commit a transaction, or all reject it. A single process, often known as the coordinator, proposes a single transaction. 2-Phase Commit (2PC) and 3-Phase Commit (3PC) are two notable protocols for distributed commit [24, 25]. 2PC works in two phases:

1. The coordinator sends a voting request to all participants along with the transaction. Each participant answers by either agreeing to the transaction, or rejecting it.

2. Once all answers are received, the coordinator sends a global-commit request if all participants agree, or a global-reject if at least one participant rejected the transaction in phase 1. Every participant must acknowledge the phase 2 request. Once all acknowledgements are received, the transaction is considered successful by the coordinator.

However, 2PC is a blocking protocol. If both the coordinator and a participant fail during the second phase, it is impossible to decide on committing or rejecting the transaction; the protocol blocks any further commit. In some applications, 2PC also cannot accept any new commit request unless the current one has succeeded. In contrast, 3-Phase Commit (3PC) introduces a third

phase, the pre-commit round, between the voting and commit steps. The third phase allows 3PC to become non-blocking, but some corner cases may lead to inconsistencies in the decision.

Process failures are an important challenge in distributed systems and can have disastrous consequences if the algorithms do not account for such events. Some typical failure models are crash-failure, where a process stops responding, crash-recovery, where a process eventually recovers (but can lose its internal state), or Byzantine-faults, where some processes are actively working to undermine the progress of the global task.

Fault-tolerant consensus. Commit is a special case of consensus: while commit focuses on a unique transaction and participants can either commit or reject it, consensus supports more than one initial proposition. The participants then reach a consensus by agreeing on one unique proposed item. When processes can fail, we often refer to the problem as fault-tolerant consensus.

Paxos and Raft are two well-known solutions for fault-tolerant consensus [26–28]. As long as a majority of processes are operational and reachable, Paxos eventually achieves consensus. Paxos assumes an asynchronous, non-Byzantine system with crash-recovery: messages can be dropped and delayed, but not tampered with; the network can be partitioned; nodes can crash and recover, and have access to persistent storage. However, Paxos requires eventual synchrony to terminate: eventually, a majority of nodes will have bounded message delays.

In Paxos, processes can take up to three different roles: **(a)** proposers propose a value to agree on, and act as coordinators for the protocol's execution. Unlike 2PC and 3PC, where at most one coordinator must be present, Paxos supports the presence of multiple proposers. **(b)** acceptors reply to proposers requests by accepting proposals. They informally act as the system's distributed memory. **(c)** learners do not participate in the consensus: they only learn the agreed value once a consensus is met.

The protocol consists of two phases: the *Prepare* phase and the *Accept* phase, and is depicted in Fig. 1.3. In an informal, high-level perspective, Paxos executes as follows:

1. During the Prepare phase, a proposer starts a consensus by contacting a majority of acceptors. By doing so, it learns if a value has already been accepted by any participant, and ensures that no older requests can go through.

2. In the Accept phase, the proposer adopts the latest accepted value if any has been accepted so far, and requests a majority of acceptors to accept this value. Once a majority has been reached, the value is considered chosen; no different value can be chosen afterwards.

1.2.3 Embedded Intelligence

With their recent advances, Artificial Intelligence (AI) and Machine Learning (ML) have established themselves as important methods for decision-making processes. First in computer vision, followed by natural language processing and speech recognition, deep neural networks were shown to outperform many traditional methods formerly considered state-of-the-art. Learning techniques

can be divided into broad categories, where some concepts overlap (e.g., semi-supervised learning) [29]. The three most common categories are:

- *Supervised learning*: a dataset of features and corresponding labels is available, the goal is to find a generalized function mapping features to labels;
- *Unsupervised learning*: an unlabeled dataset is available, the goal is to find if a hidden structure exists within; and
- *Reinforcement learning*: an environment, simulator, or traces are available, the goal is to find a sequence of interactions (a policy) leading to a desired, final state.

Deep learning, a sub-field of machine learning, is characterized by neural-network models relying on multiple consecutive layers, sometimes up to a hundred layers. Deep models therefore consist of millions of parameters (e.g., weights), that must be stored and retrieved when executing the model. For example, the ResNet101 model, used in computer vision, contains 44.7M parameters. BERT, a state-of-the-art model for many natural language processing applications, includes 110M parameters and can go up to 330M [30]. Storing such models require hundreds of Megabytes of storage up to Gigabytes, as their parameters are often using 32-bit floats. As such, deep learning often rely on cloud resources for training, and such heavy models can only be executed on powerful devices.

Tiny Machine Learning. However, we see a growing need for machine intelligence that can be executed on limited devices: for example, face recognition and augmented reality on smartphones, as well as speech recognition and classification on embedded IoT devices. The reasons to rely on an embedded execution rather than offloading computation to the cloud, are three-fold: reducing latency when communication is sporadic or throughput is limited, reducing the amount of data sent to save energy, as well as preserving privacy by not sending sensitive data at all. Standard deep models, however, are too large to be stored and executed on resource-constrained platforms. Several approaches reduce their memory footprint to enable their on-device execution: smaller models implementing efficient operations naturally require fewer parameters; for example, MobileNet achieves accuracy on-par with ResNet101 while requiring only 3.5M parameters [31]. Quantization optimizes both model size and execution time: by transforming the parameters from a floating-point representation down to an integer representation, often using an 8-bit format per weight, the model takes up less space and avoids floating-point arithmetic altogether, therefore saving energy in the process [32]. Relying on optimized operations such as in binary neural networks or by using sparse-matrices multiplication further improves on-device model executions [33, 34]. Finally, some approaches split the neural-network into chunks and offloads the computation to neighboring platforms, edge devices, or the cloud [35, 36]. The use of machine learning on limited devices is often referred to as Tiny Machine Learning (TinyML).

Reinforcement Learning. Reinforcement Learning (RL) refers to a sub-field of machine learning concerned with sequential decision processes, where learning occurs via interaction with an external environment [37]. The goal of an RL solution is to learn what to do, i.e., select the best action, in the

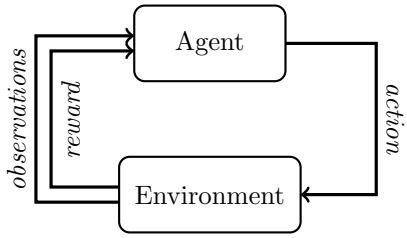

Figure 1.4. The basic Reinforcement Learning (RL) model.

current situation. More importantly, a sequence of actions is often necessary to reach a desired state. Instead of requiring a dataset comprising input and labels, RL methods rely on a reward signal quantifying how beneficial reaching a new state is, where the reward reinforces the system's knowledge of the benefits or detriments of taking a given action. In some problems, the reward is not immediate, but obtained at the end of a sequence of actions. Therefore, RL solutions maximize the reward function over time rather than the immediate reward. Formally, the agent seeks to maximize the cumulative reward $R_t \triangleq \sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau$, where r_τ is the reward obtained when transitioning at time τ, and $\gamma \in [0, 1)$ a constant called the discount factor, where a small discount factor will force the agent to maximize immediate reward while a high discount maximizes long-term rewards. Sutton and Barto summarize the three main characteristics of RL problems as follows:

1. The problem is a closed-loop system,

2. There is a lack of prior knowledge, and

3. The consequences of actions play out over extended time periods [37].

Fig. 1.4 illustrates the basic interaction in RL problems. The learning algorithm, called the agent, is surrounded by and interacts with its external environment. The agent obtains observations, i.e., input features, by observing the state of the environment, and acts by selecting and applying an action. The environment then returns both its new internal state and the reward obtained by the agent's decision. During training, the agent must carefully mix exploration and exploitation, by either choosing action randomly to discover new sequences, or by applying the best action to maximize the reward. Most RL problems can be represented as Markov Decision Processes (MDPs) [38], an extension of Markov chains for decision-making processes. While Markov chains represent state transitions as a probabilistic distribution, the transitions in MDP are affected both by the agent's decision and some randomness. Formally, an MDP is represented by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the set of possible states, $\mathcal{A}$ the set of possible actions, $\mathcal{P}$ the transition probability function and $\mathcal{R}$ a reward function. The MDP is said to be finite if $\mathcal{S}$ is finite. Most RL results suppose a finite MDP.

Q-Learning. Q-learning is a well-known method for RL problems [37]. In Q-learning, an agent learns an action-value function $Q(s, a)$ representing the expected cumulative reward the agent should get when starting in state s,

using action a such as:

$$Q(s,a) \triangleq \mathbb{E}\big[R_t \mid s_t = s, a_t = a\big] \tag{1.1}$$

If the environment can be modeled as an MDP, then we can find the optimal function $Q^*(s,a)$ that follows Bellman's principle of optimality [39]:

$$Q^*(s,a) = \mathbb{E}\big[r_t + \gamma \max_{a'} Q^*(s',a') \mid s_t = s, a_t = a\big] \tag{1.2}$$

where r_t is the immediate reward received, γ the discount factor, and s' the state reached at time $t+1$. By iteratively trying actions and receiving rewards, we can update a Q-function that ultimately converges to the optimal $Q^*(s,a)$, i.e., we learn the optimal sequence of actions that maximizes the reward.

While Q-learning algorithms have historically used a tabular approach in estimating the Q-function [37], Mnih et al. demonstrated that deep neural networks can act as Q-function approximators [40]. Deep Q-networks (DQN) have been successfully used to solve various problems, for example datacenter cooling [41], wireless modulation [42], and CSMA/CA optimization [43]. The advantage of DQN over tabular approaches is the ability to solve problems with continuous states and the generalization property of neural networks.

1.3 Related Work

In this section, we present a curated selection of works that represent the current state of the literature in the field of this thesis. We categorize related work in the following aspects: **(a)** the intersection of distributed systems and wireless networking, with a focus on providing consistency in low-power IoT, **(b)** the optimization of adaptive wireless protocols, detecting external interference, and the use of AI for wireless networking, and **(c)** methods for environment detection and the applications of wireless fingerprinting.

1.3.1 Wireless Distributed Systems

We first provide a survey of concepts of distributed systems applied to low-power wireless communication, with topics ranging from failure detection, reliable communication, and consensus.

Failure detection. In a wireless deployment, an unresponsive node can be caused either by a node failure, (temporary) communication failures due to, e.g., interference, or a node moving out of range due to mobility. Detecting the cause of unresponsiveness is non-trivial, as communication faults and node failures are sometimes hard to distinguish [44]. Ruiz et al. propose MANNA, a self-diagnostic and self-healing solution to fault management using active requests [45]. Miao et al. use correlation patterns to detect possible silent faults in wireless sensor networks [46]. Jhumka and Mottola combine theoretical and systems approaches to tackle the problem of neighborhood view consistency, i.e., the accuracy and correctness of localized neighbor information [47, 48]. While they prove that it is impossible to design a localized solution to strong neighborhood view consistency, they propose a localized algorithm for weak view consistency. By aggregating the 2-hop neighbor information, their solution

raises signals whenever a transient fault or node crash has occurred in the vicinity.

Message delivery. In wireless networks, the communication substrate is often subject to faults, particularly message delivery failures. Several reliable transport protocols, some sharing similarities with TCP, have been proposed to tackle the problem of guaranteeing message delivery. To provide reliable data acquisition for a structural monitoring deployment, Xu et al. propose Wisden, a data transport protocol combining end-to-end as well as hop-by-hop recovery mechanisms [49]. Paek et al. go further with RCRT and incorporate congestion control in addition to reliable delivery in low-power wireless networks [50]. Kumar et al. endowed TCP with several optimizations to fit it to resource-constrained wireless nodes [51]. Building on synchronous transmissions and LWB [52], VIRTUS brings virtual synchrony to low-power wireless networks [53]. Virtual synchrony provides reliable atomic multicast, i.e., VIRTUS guarantees the message delivery to all recipients of the multicast, while guaranteeing that messages are delivered in order. Time-critical cyber-physical systems, such as industrial control systems, have stringent requirements in terms of latency but often can survive few message losses [54]. Thus, reliable data delivery is not imperative in such deployments. Instead, bounded-delivery protocols fulfill the requested time requirements. Chipara et al. propose a centralized deadline-based scheduler to ensure on-time delivery [55]. Li et al. extend it for emergency alarms over wireless [56]. With TTW, Jacob et al. show that wireless solutions can replace wired field buses in industrial settings [54]. By building on synchronous transmissions, TTW provides end-to-time timing predictability, high reliability, and low latency.

Consensus. Due to the multi-hop and mobile nature of some wireless deployments, distributed wireless consensus require novel solutions often tailored to the specificities of the network architecture. Benchi et al. use epidemic routing and the One Third Rule to achieve agreement in opportunistic networks [57]. Chockler et al. augment nodes with collision detectors to enable fault-tolerant consensus in single-hop ad-hoc deployments [58], while Turquois further allows consensus in the presence of Byzantine faults in single-hop networks [59]. With JAG, Boano et al. take full advantage of the destructive behavior of jamming signals, as JAG uses wireless jamming to acknowledge one-hop agreement requests [60]. Köpke investigates the performance of 2-Phase Commit (2PC) for wireless sensor networks [61]. The author shows that both low-latency and high-reliability MAC and routing layers are necessary to provide commit semantics with adequate performance in sensor networks. With RedMAC and the routing protocol Net, Köpke provides a wireless stack able to run the original 2PC for 16 participants under 2 min. However, the author only hints at additional improvements, such as multicast and aggregation, but does not provide a dedicated 2PC design tailored to wireless communication. In contrast, Borran et al. extends Paxos with a new communication layer for 802.11 opportunistic networks [62]. The authors build a tree to route and collect acceptor responses.

Building on top of synchronous transmissions and reusing concepts from Chaos [20], Agreement in the Air (A^2) introduces 2-Phase Commit (2PC) and 3-Phase Commit (3PC) to low-power lossy networks, providing commit guarantees with low-latency and high reliability [63]. Later, Spina et al. propose

a new approach to 2&3PC named XPC [64], combining Chaos with Glossy floods. Compared to A^2, their approach has the advantage of terminating faster during aborts while providing similar latencies for successful commits. After the publication of Paper A of this thesis (see Chapter 2), Spina worked on a different Paxos implementation, reusing the approach used in XPC, called WISP [65]. The author obtained performance of the same order as the one provided in our Paper A for Paxos but higher latency variation in their Multi-Paxos implementation.

Consensus algorithms also play an important part in the design of blockchain technologies, and several works investigate wireless consensus in the context of blockchains operating on IoT devices. wChain relies on aggregation by appending all messages received from lower-tier followers to the next transmission to speed up dissemination, three phases, and a quorum of responders to lead the agreement [66]. Xu et al. implement Raft, a (non-Byzantine) fault-tolerant consensus algorithm similar to Paxos, to drive consensus for a private blockchain distributed over IoT devices. By relying on the majority-approach and fault-tolerance of Raft, the authors support malicious jamming affecting parts of the wireless medium [67]. SENATE provides Byzantine and Sybil fault-tolerant consensus for multi-hop wireless networks [68]. By relying on a selfish ALOHA competition to select a quorum, SENATE ensures that malicious nodes cannot represent a majority of the participants and further execute a byzantine agreement to drive the consensus. The authors use the example of agreement at a road intersection to evaluate their protocol.

Vehicle coordination. Vehicle-to-vehicle communication usually relies on cellular technologies, e.g., LTE or 5G, or IEEE 802.11p, to communicate [69]. In contrast, in Chapter 3, we argue that using low-power radios allows more participants, e.g., bikes and pedestrians, to coordinate. In the case of a road intersection, a coordination protocol must ensure the safety of its participants while minimizing the delay a vehicle or pedestrian must observe before being able to cross. Dresner and Stone propose AIM, a centralized intersection management protocol relying on the cellular infrastructure [70]. Ferreira et al. propose Virtual Traffic Lights, a decentralized solution where cars elect a leader mimicking traffic lights to control the intersection [71]. In contrast, we propose in Chapter 3 a leader-based solution using commit semantics, where access to the intersection is based on the waiting time of the participating vehicles.

1.3.2 Adaptive Low-Power Wireless

An adaptive system is a system able to detect internal or external changes and able to modify its behavior to counteract the effects of such disturbances. In this thesis, we use adaptivity to refer to reaction to external adverse effects on the wireless medium, e.g., wireless interference. The term can however have a broader sense in the literature, e.g., adapting to new internal data traffic.

Detecting changes. Collision avoidance through channel assessment is a common approach to adapt to the medium in wireless communication and is standard in IEEE 802.15.4 [6], although channel assessment can only detect interference on the sender-side. Several works propose metrics to measure and quantify link quality, and its evolution through time. Srinivasan et al. as well as

Munir et al. propose to use link burstiness as a metric for routing [72,73]. Noda et al. propose the Channel Quality metric as a measure of channel availability over time [74]. MUSTER additionally relies on available energy at relays to enable energy-efficient routing [75]. To adapt to internal data traffic changes, the Low-power Wireless Bus (LWB) centrally schedules communication and adjusts its round interval and slot attribution [52]. Blink builds on top of LWB and goes a step beyond, providing delivery guarantees to deadline-based flows while maintaining adaptivity to changing traffic demands [76]. Because LWB, and thus Blink, are based on network-wide Glossy floods [9], they are impervious to node mobility and, to some extent, link quality. However, we show in our Paper C (see Chapter 4), that interference nonetheless degrades the performance of LWB.

Resource management in BLE. Spörk et al. propose a channel-exclusion mechanism to improve Bluetooth Low Energy communication in the presence of interference by relying on the per-channel Packet Delivery Ratio [77]. Mast et al. extends the idea with channel re-inclusion, by re-including channels after static timeouts [78]. Independently of Mast et al., we introduce in Chapter 5 eAFH, a channel-management mechanism for BLE able to both dynamically deactivate and re-include channels via exploration and dynamic timeouts. We show that exponentially increasing timeouts are better equipped to deal with varying interference and that eAFH increases channel diversity by more than 40% compared to state-of-the-art approaches. Other works propose to modify the Channel Selection Algorithm (CSA) used by BLE to better avoid interfered channels: Pang et al. introduce the Interference Awareness Scheme (IAS) featuring a new CSA weighing the channels by the probability that interference is present [79]. Cheikh et al. introduce SAFH and also rely on weights to improve channel selection [80]; channels with low frame error rates have a lower probability to be used next or are excluded altogether, while good channels are more likely to be used. However, modifying the CSA is more cumbersome than excluding channels. The CSA must be implemented both by the central and peripheral, while only the central is required to implement the exclusion strategy. Park et al. improve energy consumption and QoS in BLE with AdaptaBLE [81], by controlling the transmission power, BLE physical mode, and connection interval. BLEX modifies the communication scheduling behavior of the Zephyr RTOS to adapt to different Quality of Service requirements [82].

AI for networking. A recent trend employs machine learning, and especially Reinforcement Learning, to learn optimal parameters in wireless protocols and to enable adaptive behaviors. At the physical layer, Vrieze et al. set out to entirely learn a modulation scheme [42]. Using a known preamble, policy-gradient methods, and two independent agents, their system iteratively tries to modulate and demodulate the transmitted preamble over a noisy channel, until both agents can reconstruct the message. The agents converge to a rotated version of 16-QAM, without any prior knowledge of modulation techniques. At the data link layer, Amuru et al. as well as Mastronarde et al. learn contention for 802.11 CSMA/CA, both using post-decision state-based learning [43,83]. Meyer and Turau use tabular Q-Learning to identify hidden traffic patterns for Guaranteed Time Slots allocation in CSMA/CA [84]. Kwon et al. use multi-agent deep RL to learn transmission power, therefore controlling the

communication range to optimize energy [85]. Savaglio et al. introduce QL-MAC, a duty-cycled protocol adapting in sleep and active periods to minimize energy consumption without impacting communication and able to react to data traffic changes [86]. Similarly, Trinh et al. rely on Q-Learning to optimize duty-cycling [87].

Dakdouk et al. propose a channel selection scheme for IEEE 802.15.4-TSCH using multi-armed bandits [88]. Zhang et al. use multi-armed bandits to optimize Glossy floods [89]. In their work, each IoT platform runs Exp3 independently, where an arm represents the number of retransmission in a flood. In Chapter 4, we also tackle the problem of adaptivity in glossy-based communication. While Zhang et al. focus on static environments where interference does not evolve and try to find the optimal per-node retransmission parameter, we instead look at adaptivity in the presence of intermittent interference. We combine a centralized deep Q-Learning mechanism to react to interference with distributed multi-armed bandits to optimize energy in normal conditions.

Similar to our argumentation in this thesis, Restuccia and Melodia argue that deep-RL can be used to reconfigure the wireless stack to adapt to wireless changes [90]. They propose DeepWiERL, a hardware-software framework to execute and train DRL on IoT platforms. They combine computation over FPGAs, transfer learning, and deep reinforcement learning to create self-adaptive behaviors. Joseph et al. go a step further and argue for self-driving radios, a paradigm where the wireless stack learns its optimal configuration from high-level only specifications of the scenario [91]. However, they simply create and train a new DQN whenever a new scenario is defined, which is memory-intensive and requires a training environment at hand. In contrast, we argue that training a general RL agents able to control the parameters of the network stack should suffice to cover different application scenarios.

1.3.3 Environment Detection and Wireless Fingerprinting

Wireless signals are not limited to the sole exchange of information over distance. A significant body of work demonstrates that the characteristics of wireless signals form unique fingerprints, that allow the accurate position estimation of a device, detecting and locating a person within a room, or estimating the activity and movement of persons. In this section, we present several results building upon wireless fingerprints.

RF localization. The wireless medium is a complex environment whose characteristics evolve with time and location. The objects surrounding a device equipped with a wireless radio, e.g., walls, doors, and windows, cause signal reflections and attenuation. If the location of specific transmitting devices, known as anchors, is given, it is possible to estimate the location (i.e., precise position) of a receiver without the need for an energy-hungry GPS. Typical approaches combine signal attenuation, time of arrival, time difference of arrival, and angle of arrival to estimate the position of a specific device within, e.g., a room. However, indoor environments complicate the localization due to multi-path fading and obstacles. Bahl and Padmanabhan introduce RADAR, an RF-based indoor localization system via triangulation [92]. The authors use the K-Nearest Neighbors, empirical measurements, and propagation models to estimate a user position and track moving targets within an office environment.

Giorgetti et al. propose relative localization via self-organized maps [93]: a device estimates its relative position within a network in the absence of anchors. Yang et al. combine the fingerprint map, i.e., a map containing RSS measurements at different locations as in RADAR, with the distance walked between two measurements to create a high-dimensional fingerprint space and a stress-free map. They then use a nearest-neighbor approach to find the nearest location at query time. More coarse-grained, Chow et al. perform locality classification, i.e., they classify the user position within pre-defined sections of the area of interest [94]. Li et al. review potential attacks on localization and propose statistical methods to mitigate their adverse effects [95]. Among others, jamming, line-of-sight, and signal strength attenuation, replay attacks and routing path alteration are listed as possible attacks on such algorithms. Deep learning has also been demonstrated to perform well in RF localization [96]. Several works investigate the use of Bluetooth and BLE packets as a driver for localization and distance estimation [97, 98]. Bertuletti et al. demonstrate that RSS measures are noisy and lead to a 30% distance estimation error [97], while Zhuang et al. achieve <3 meters localization using BLE advertisements [98].

Device fingerprinting. The wireless medium is not the only component molding a signal with identifiable elements. Wireless radios also imprint unique characteristics onto physical transmissions due to imperfections in their circuitry. For example, modulation shifts, self-interference, amplitude clipping, and frequency offset are possible impairments caused by the wireless radio of a device, and are different even for two devices sold by a manufacturer as identical. With such knowledge, it is possible to identify the sender of a transmission from its unique fingerprint. Such identification can be used, for example, as a replacement for computation-intensive cryptography for device authentication. Brik et al. introduce PARADIS, a system identifying wireless senders via support vector machines and K-nearest neighbors [99]. Nguyen et al. rely instead on unsupervised learning and Gaussian mixture models, creating clusters of fingerprints and checking whether a MAC address is used by more than one device, therefore detecting communication attacks [100]. Peng et al. show that device fingerprints are consistent over long periods of time by classifying devices 18 months after collecting data and training their classifier [101]. Sankhe et al. present ORACLE, a highly accurate CNN-based device classification system [102]. However, Al-Shawabka et al. demonstrate that changing wireless conditions can greatly impact the performance of CNNs [103].

Device-free localization. Instead of locating a wireless device within an environment, wireless perturbations can also be used to detect a person or object of interest within an environment, even if such person does not wear a wireless transceiver; we then speak of device-free localization. Zhang et al. show how received signal perturbations within a grid deployment can estimate the position of a person [104]. SCPL is both able to count and localize multiple subjects using device-free localization [105], by first iteratively counting and canceling the impact of a single subject until no perturbations persist, and later quantifying the RSS perturbations within cells to localize targets. Mager et al. evaluate the effect of altered environments, e.g., when furniture is moved, over the performance of device-free localization and present mechanisms to mitigate such effects [106].

Activity detection. Not only do wireless perturbations allow the detection

and location estimation of device-free humans, but their variations over time can also be used to estimate the current activity of the target. Sigg et al. execute device-free activity recognition and distinguish between lying, standing, walking, and crawling at several locations by deploying one transmitter and one receiver in a corridor [107]. Sigg et al. also show that the RSSI of WiFi transmissions received by a phone are sufficient to detect the presence and activity of humans nearby the device [108]. In CARM, Wang et al. rely instead on the CSI and combine features extracted from multiple receivers to distinguish between eight activities [109]. Wang et al. further combine localization and activity recognition in one system and introduce auto-encoders to automatically extract informative features from the wireless signals [110]. Apart from wireless fingerprinting, microphones, cameras, accelerometers, and sometimes even barometers, are common instruments to execute Human Activity Recognition (HAR) [111–113].

Environment detection. While the localization problem focuses on accurate position estimation, environment detection instead aims at categorizing the type of environment into general classes such as home, office, street, or shop. Several works establish acoustic sensing as an accurate enabler for environment detection. Ma et al. rely on microphones and Hidden Markov Model classifiers to distinguish between 12 environments such as bus, car, street, and office, from 3-second long audio recordings and achieve up to 93% accuracy [114]. However, the authors do not discuss the problems of privacy arising from relying on microphones to infer surroundings. Heittola et al. represent audio fingerprints as histograms and compare new recordings with previous histograms to distinguish between ten environments [115]. Choi et al. combine microphone and camera inputs to detect the user position and activity [116]. Liang and Wang introduce a CNN to parse a smartphone's accelerometer data and distinguish which transport (such as bus, car, bike) the user is using [117]. In this thesis, we argue in Chapter 6 that Bluetooth Low Energy advertisements received by a device are sufficient to classify its surrounding environments, and are less intrusive than audio-based systems due to the randomized addresses used by BLE advertisers.

1.4 Research Statement and Contributions

Within the Internet of Things ecosystem, low-power wireless IoT devices stand out for their reliance on resource-limited hardware, their stringent energy limitations, and their use of unreliable wireless communication. Nonetheless, we expect low-power IoT systems to support a broad set of applications, ranging from swarms of drone control, wireless closed-loop industrial systems, safety-critical monitoring infrastructures, and deep learning inference. Such applications will require IoT devices to communicate, coordinate, and collaborate to reach their goals. Further, as the number of IoT deployments grows, so does the use of the wireless resources and overall data traffic. IoT systems must be able to detect and react to changes in their wireless environment to deal with highly dynamic and unpredictable external changes.

Collaborative IoT. Although many wireless standards and protocols have reached a mature state and can provide communication with high packet delivery, many lack the theoretical guarantees required to enable safe and

consistent coordination between devices. In particular, wireless IoT systems lack efficient solutions to the problem of fault-tolerant consensus, where devices coordinate over an unreliable wireless medium. Consequently, this thesis first focuses on answering the following question:

RQ1: *How can we provide low-latency coordination mechanisms with consistency guarantees for low-power wireless IoT, where connectivity is unreliable and devices can fail?*

We provide our answer to this question in Papers A and B. Paper A introduces a general communication primitive providing fault-tolerant consensus, while Paper B focuses on the application of vehicular coordination at intersections with safety guarantees.

Adaptive IoT. Due to the ever-growing number of IoT systems deployed, we see an overcrowding of the wireless spectrum. Concurrent wireless traffic is becoming increasingly difficult to predict and provision against. In parallel, some IoT devices face mobility challenges, where they must adapt their behavior to their current surroundings. IoT systems require detection mechanisms able to recognize their environments and identify changes within. For example, the network stack used by such devices should autonomously react to external perturbations to counteract their effects on the overall performance of the system; while wireless headphones should adapt noise cancellation based on the environment. We formulate the challenge of environment identification and autonomous reaction in two questions and answer them in the last part of this thesis:

RQ2: *How can we design methods able to detect changes to the wireless medium and react to them to maintain key performance metrics?*

RQ3: *How can resource-limited devices accurately recognize their environment solely from low-power sensors?*

Papers C and D answer the question of autonomous detection and reaction in two different wireless standards, while Paper E focuses on environment detection. Paper C introduces an adaptive communication primitive relying on reinforcement learning to detect and react to external wireless interference. Paper D proposes a channel-management system for Bluetooth Low Energy with a special focus in dynamic environments and mobile scenarios. Finally, Paper E demonstrates that it is possible to distinguish between different environments such as homes, offices, or shops, solely from Bluetooth Low Energy signals fed into an embedded ML model.

Contributions

Paper A - Paxos Made Wireless: Consensus in the Air

This paper addresses the problem of fault-tolerant consensus in low-power wireless networks. While consensus is a mature field of research in wired networks, we argue that typical solutions of the wired domain do not satisfy the performance requirements of resource-constrained wireless networking. We introduce *Wireless Paxos*, a new flavor of Paxos fitted to the characteristics of

low-power wireless networking: we show that Paxos can be transformed from a unicast scheme to a many-to-many scheme, which can be efficiently executed in low-power wireless networks. We co-design the consensus algorithm along with the lower layers of the network stack to greatly improve the latency of consensus and have tighter control on the transmission policy. The overall result is a broadcast-driven consensus primitive using in-network processing to compute intermediate results in Paxos. Our solution builds on top of Synchrotron [63], a kernel for synchronous transmissions inspired by Chaos [20], providing a basis for highly reliable and low-latency networking in low-power wireless with support for in-network processing. Our results show that Wireless Paxos requires only 289 ms to complete a consensus between 188 nodes in testbed experiments. Furthermore, we show that Wireless Paxos stays consistent even when injecting node failures.

Personal contribution. I am the main author and main designer of Wireless Paxos[1]. The chapter was published as a paper at the International Conference on Embedded Wireless Systems and Networks (EWSN), 2019 [118], and was nominated as *candidate for the best paper award* at the conference.

Paper B - STARC: Low-power Decentralized Coordination Primitive for Vehicular Ad-hoc Networks

This paper revisits the coordination problem with a focus on vehicle-to-vehicle (V2V) communication. V2V communication is expected to improve road usage efficiency through cooperative driving, platooning, and autonomous intersection management by replacing traffic lights with digital counterparts. However, such V2V coordination protocols must ensure the safety of all road users by guaranteeing consistency among participants and perform timely even under communication failures. We introduce STARC, a decentralized reservation-based protocol that uses low-power wireless radios to enable energy-efficient vehicle-to-vehicle communication. We build our coordination protocol on top of Synchrotron, a low-latency and energy-efficient communication primitive for all-to-all communication [63]. With STARC, traffic participants reserve lanes to cross intersections. Our mechanism allows users to reserve lanes through an intersection by providing transaction semantics and distributed commits. As a result, vehicles have unique access to different parts, i.e., lanes, of the intersection, ensuring that at most, one car can use a given lane. We show that STARC reduces average waiting times by up to 50% compared to a fixed traffic light schedule in traffic volumes with less than 1000 vehicles per hour. Moreover, we show that the protocol supports dynamic priority strategies and we illustrate a platoon extension that allows STARC to outperform traffic lights even with traffic loads surpassing 1000 vehicles per hour.

Personal contribution. I am the second author and a co-designer of STARC[2]. The chapter was published as a workshop paper at the International Workshop on Intelligent Transportation and Connected Vehicles Technologies (ITCVT) at the Network Operations and Management Symposium (NOMS), 2020 [119].

[1] Available at github.com/iot-chalmers/wireless-paxos
[2] Available at github.com/ds-kiel/starc

Paper C - Dimmer: Self-Adaptive Network-Wide Flooding with Reinforcement Learning

In this paper, we observe that many low-power protocols are invariant to their environment dynamics and deal with interference through over-provisioning. Instead, we argue that low-power wireless networking should adapt to the wireless medium to meet a target performance, even under varying conditions, while still ensuring energy efficiency. We introduce Dimmer as a self-adaptive, all-to-all communication primitive. Dimmer builds on top of LWB [52] and uses deep Reinforcement Learning to tune its flooding parameters to match the current properties of the wireless medium. By learning how to behave from unlabeled traces, Dimmer adapts to different interference types and patterns, and is even able to tackle previously unseen interference. We evaluate our protocol on two deployments of resource-constrained nodes and show that Dimmer outperforms baselines such as non-adaptive ST protocols (∼27%) and PID controllers, and show a performance close to hand-crafted and more sophisticated solutions, such as Crystal (∼99%).

Personal contribution. I am the main author and main designer of Dimmer[3]. The chapter was published as a paper at the IEEE International Conference on Distributed Computing Systems (ICDCS), 2021 [120].

Paper D - eAFH: Informed Exploration for Adaptive Frequency Hopping in Bluetooth Low Energy

This paper tackles the problem of frequency management in the context of dynamic environments and mobility. Due to the ever-growing traffic in the already crowded 2.4 GHz band, the quality of Bluetooth Low Energy connections drastically varies with nearby wireless traffic, location, and time of day. These dynamic environments demand new approaches for channel management by both dynamically excluding frequencies suffering from localized interference and adaptively re-including channels, thus providing sufficient channel diversity to survive the rise of new interference. We introduce eAFH, a new channel-management approach in BLE with a strong focus on efficient channel re-inclusion for mobile scenarios. eAFH introduces informed exploration as a driver for inclusion: using only past measurements, eAFH assesses which frequencies we are most likely to benefit from re-inclusion into the hopping sequence. As a result, eAFH adapts in dynamic scenarios where interference varies over time. We show that eAFH achieves 98-99.5% link-layer reliability in the presence of dynamic WiFi interference with 1% control overhead and 40% higher channel diversity than state-of-the-art approaches.

Personal contribution. I am the main author and main designer of eAFH[4]. The chapter was published as a paper in the Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS), 2022 [121].

[3] Available at github.com/ds-kiel/dimmer
[4] Available at github.com/ds-kiel/eAFH

Paper E - BlueSeer: AI-Driven Environment Detection via BLE Scans

In this paper, we argue and demonstrate that it is feasible to infer an environment, i.e., we can distinguish between a home, an office, or a street, solely from packets received with a Bluetooth Low Energy radio. IoT devices rely on environment detection to trigger specific actions, e.g., for headphones to adapt noise cancellation to their surroundings. While phones feature many sensors, from GNSS to cameras, small wearables must rely on the few energy-efficient components they already incorporate. In this paper, we demonstrate that a Bluetooth radio is the only component required to accurately classify environments and present BlueSeer, an environment-detection system that solely relies on received BLE packets and an embedded neural network. BlueSeer achieves an accuracy of up to 84% differentiating between 7 environments on resource-constrained devices and requires only $\sim$ 12 ms for inference on a 64 MHz microcontroller unit.

Personal contribution. I am the first author and a co-designer of BlueSeer[5]. The chapter was published as a paper in the Proceedings of the ACM/IEEE Design Automation Conference (DAC), 2022 [122], and was nominated as *candidate for the best paper award* at the conference.

1.5 Conclusions and Emerging Directions

In this thesis, we argue that the next generation of low-power IoT devices will communicate, coordinate, and collaborate to execute distributed applications of ever-increasing complexity in challenging and dynamic environments. Future IoT systems will also exhibit adaptive behaviors and react to external events to optimize their performance. We identify two key challenges such systems must face to reach these goals: (1) low-power IoT devices will require efficient coordination mechanisms providing consistency and safety guarantees, and (2) they will need methods to recognize their external environments, detect changes within, and react to them to maintain key performance metrics.

This thesis presents five key mechanisms to facilitate collaboration and adaptivity in future IoT systems. To enable efficient coordination, we show that solutions of the wired domain can be adapted and coupled to low-level wireless communication primitives, therefore ensuring low energy consumption and low latency while maintaining their consistency guarantees even in the presence of unreliable transmissions and participant failures. To endow IoT devices with adaptive behaviors, we introduce methods able to distinguish environments solely from wireless signals, detect external changes to the wireless medium, and adapt to possibly unseen events in mobile and dynamic scenarios. Particularly, we demonstrate the practicality of embedded machine learning in decision-making processes on resource-limited platforms.

We identify three directions that we believe future IoT systems can benefit from. Future IoT deployments should be self-forming and depict self-adaptive behaviors: they should coordinate to find the optimal communication strategies or parameters, and provide consistency guarantees in their configuration.

[5] Available at github.com/ds-kiel/blueseer

Secondly, embedded machine learning, especially in the form of reinforcement learning, is a promising approach to enable decision-making on limited hardware, as it allows devices to find the optimal strategy even if such a strategy is unknown to the designer. We argue that an AI-enabled wireless stack would be able to optimally adapt to dynamic and challenging environments, even in previously unseen conditions. Finally, while we provide consistent coordination in the presence of network and device failures, we assume the absence of adverse processes actively undermining the coordination process via malicious communication, i.e., we assume no Byzantine failures. As many multi-hop IoT deployments rely on devices to relay messages, Byzantine-fault tolerant mechanisms that are practical in real deployments, i.e., efficient in terms of message exchanges and energy consumed, are required in safety-critical applications where adversarial actors can add devices to a system.