Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Ingegneria Elettronica, Telecomunicazioni e Tecnologie dell'Informazione

Ciclo 35

Settore Concorsuale: 09/F2 - TELECOMUNICAZIONI

Settore Scientifico Disciplinare: ING-INF/03 - TELECOMUNICAZIONI

PROXIMITY DETECTION PROTOCOLS FOR IOT DEVICES

Presentata da: Elia Leoni

Coordinatore Dottorato

Aldo Romani

Supervisore

Amy L. Murphy

Co-supervisore

Elisabetta Farella

Esame finale anno 2023

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Abstract

DEI - Department of Electrical, Electronic and Information Engineering
"Guglielmo Marconi"
Electronics, Telecommunications, and Information Technologies Engineering

Doctor of Philosophy

Proximity Detection Protocols for IoT Devices

by Elia Leoni

In recent years, we have witnessed the growth of the Internet of Things paradigm, with its increased pervasiveness in our everyday lives. The possible applications are diverse: from a smartwatch able to measure heartbeat and communicate it to the cloud, to the device that triggers an event when we approach an exhibit in a museum. Present in many of these applications is the *Proximity Detection* task: for instance the heartbeat could be measured only when the wearer is near to a well defined location for medical purposes or the touristic attraction must be triggered only if someone is very close to it. Indeed, the ability of an IoT device to sense the presence of other devices nearby and calculate the distance to them can be considered the cornerstone of various applications, motivating research on this fundamental topic. The energy constraints of the IoT devices are often in contrast with the needs of continuous operations to sense the environment and to achieve high accurate distance measurements from the neighbors, thus making the design of Proximity Detection protocols a challenging task.

In this thesis, we addressed the challenges of proximity detection, showing how we faced them through the design of two proximity detection protocols using low-power radio technologies. With the Janus protocol we demonstrate that by merging BLE and UWB, it is possible to develop a protocol that is both accurate and energy efficient. Here the BLE radio is used for the neighbor discovery, while the UWB is triggered on demand for ranging. By doing so, we illustrate through several experiments how the protocols achieves high accuracy, while managing consumption. We used Janus in a real scenario where the joint collaboration with a data analysts research group and the high accuracy of the protocol enabled an in-depth analysis, demonstrating the flexibility of Janus and the opportunities offered by proximity detection. Subsequently, with the Hermes protocol we tackled one of the limitations of Janus, namely the lag between discovery and ranging, showing how we can achieve proximity detection using only the UWB radio and simultaneously keeping in mind the low-power constraints of IoT. For this protocol we employ a custom simulator that helps rapidly testing different configurations, showing the achievable performance. Furthermore, Hermes has been implemented on real devices, and the experimental results we achieved are in line with the simulator outcomes.

Acknowledgements

Here we are for the acknowledgments. At the end of this path, I probably have to thank so many people, that helped me to reach the final target of the Ph.D., which is physically represented by this document.

Firstly, I need to thank my supervisor Amy L. Murphy. She helped and sustained me during these crazy three years. From her I understood what being a truly passionate researcher is and how everyone should approach this type of work. I appreciated collaborating with her and I'll try to remember all the teachings that she gave to me.

Another person that definitely deserves a mention is my co-supervisor Elisabetta Farella. She is a real leader, capable of creating a fantastic environment to work in, the E3DA group. I really thank her, for the possibility that she gave me to work in FBK and spend these three years in her research group. Working together with the people of E3DA was a great experience.

I want to thank Velu Prabhakar Kumaravel, a friend always available when one needs to talk. Even if his spiritual way of thinking sometimes was in contrast with my atheism, talking with him was always interesting.

I must thank also Alberto Ancilotto. He knows everything about embedded systems, helping me several times when I had doubts. Moreover, he shares with me his passion for the role-playing game...I would say that these reasons are enough to be present here.

I also thank Francesco Paissan. Even if he talks too much about work (and deep learning), is nice to have such a passionate and interesting young researcher in the lab. He is here mostly because he is always available for a coffee break, and to go to lunch at 12 PM!

Finally, all the people at the University of Trento with who I have the privilege to collaborate: Timofei Istomin, Davide Molteni, Davide Vecchia, and Gian Pietro Picco. Has been a real pleasure to work with them. They are all fantastic researchers.

Before closing, I thank Davide Giovanelli, Gianmarco Cerutti, and Enrica Loria. Starting the Ph.D. with them around was great.

I close the acknowledgments by thanking all my family. They allowed me to attend the University, thus permitting me to start this experience that surely will give me a lot of satisfaction for my future career.

I would say that's enough. Now it's getting late, therefore *"ÀM VÀG A LÈT!"*[1]

[1] Literally, "I go to bed", in Romagnolo, my favourite language in the world.

Contents

Abstract iii

Acknowledgements v

1 Proximity Detection: a Key Application for IoT 1
1.1 *Who* is Around and at *Which Distance* . . . 1
1.2 Technologies for Proximity Detection . . . 2
1.3 Challenges for Proximity Detection Protocols . . . 3
1.4 Thesis Statement and Structure . . . 4

2 Janus: Accurate *and* Energy Efficient Proximity Detection Protocol 7
2.1 Introduction . . . 7
2.1.1 Goals, Methodology, and Contributions . . . 7
2.1.2 Requirements . . . 8
2.2 Janus Protocol Description . . . 10
2.2.1 Dual-radio Discovery and Ranging . . . 10
2.2.2 Neighbor Discovery . . . 10
2.2.3 Ranging . . . 11
2.2.4 Coordinating Discovery and Ranging . . . 11
2.2.5 Time Synchronization . . . 12
2.2.6 Assigning the Node Index . . . 12
2.2.7 Open system: Resolving Index Conflicts . . . 13
2.3 Mathematical Model . . . 13
2.3.1 Modeling the Success of Proximity Detection . . . 13
2.3.2 Overall Probability of Success . . . 13
2.3.3 Probability of Successful Ranging . . . 14
2.4 Practical Implementation . . . 15
2.4.1 Implementation Details . . . 15
2.5 From a Prototype to a Full-fledged System . . . 16
2.5.1 A Versatile Firmware . . . 16
2.5.2 A Custom Tag . . . 17
2.5.3 A Complete Solution . . . 17
2.6 In-Vitro Experiments . . . 17
2.6.1 System Evaluation . . . 17
2.6.2 Overall System Configuration and Application Scenarios . . . 17
Application Requirements and Janus Configuration . . . 17
Radio Configuration . . . 18
2.6.3 Latency and Reliability . . . 18
Experimental Setup . . . 18
Latency . . . 19
Reliability . . . 21
2.6.4 Ranging Accuracy . . . 22
Experimental Setup . . . 22
Results . . . 25
2.6.5 Energy Consumption . . . 28
2.7 In-Field Experiments . . . 29
2.7.1 Janus in Action: Experiences with COVID-19 Social Distancing 29
2.7.2 Before the Experiments: Configuring the BLE TX Power . . . 30
2.7.3 Cafeteria: Comparing BLE vs. UWB Raw Data . . . 31
2.7.4 Same-office Co-workers: Exploiting Raw Data . . . 31

2.7.5 Company-wide: Using a Higher-level Contact Definition . . . 32
2.7.6 Factory Floor: Real-time Alerting and Contact Tracing 33
2.8 Discussion . . . 34
Dual Radio Architectures - A Comparison . . . 35
2.9 Concluding Remarks . . . 35

3 Measuring Close Proximity Interactions with Janus **37**
3.1 Introduction . . . 37
3.1.1 Modeling the spread of infectious diseases from proximity data . . . 39
3.2 Materials and Methods . . . 39
3.2.1 Data acquisition . . . 39
Device setup and experimental setting . . . 41
Pre-processing of the data . . . 41
3.2.2 Definition of close proximity contacts . . . 42
3.3 Results . . . 44
3.3.1 Identification of contagion risk levels . . . 44
3.3.2 Contagion risk analysis . . . 45
3.4 Discussion . . . 48
3.4.1 Social bubbles and roles . . . 49
3.4.2 Activity Type . . . 49
3.4.3 Limitations . . . 52
3.5 Concluding Remarks . . . 52

4 Hermes: Low-Latency Proximity Detection **55**
4.1 Hermes In a Nutshell . . . 55
4.2 Simulation . . . 57
4.2.1 Simulator Description . . . 57
4.2.2 Simulation Results . . . 59
Understanding Discovery Rate . . . 59
Evaluating Discovery Rates . . . 60
Evaluating Energy Trade-off . . . 60
Evaluating Discovery across Multiple Epochs . . . 62
4.3 Implementation . . . 64
4.3.1 Hermes Configuration Details . . . 64
4.3.2 Energy Saving Techniques: Deep Sleep and Preamble Sniffing . . . 64
4.3.3 Experimental Results . . . 65
Reliability . . . 65
Energy Consumption . . . 66
Preamble Sniffing Evaluation . . . 68
4.3.4 An Eye to the Future: From the DW1000 to the DW3000 . . . 69
4.4 Concluding Remarks . . . 70

5 Validation of Indoor Positioning Solution for Automotive **71**
5.1 Introduction . . . 71
5.1.1 Considered Scenario . . . 72
5.2 System Description . . . 72
5.2.1 System Architecture Overview . . . 73
5.2.2 UWB Evaluation System Setup . . . 74
UWB Anchor Installation . . . 75
Time Synchronization . . . 75
Antenna Offset . . . 75
Coordinate Conversion . . . 76
Protocol Details . . . 77
5.2.3 UWB Accuracy Evaluation in the Garage . . . 78
Evaluation . . . 79
5.2.4 Final UWB Anchor Locations in Garage . . . 80

5.3 Experiments and Results 81
5.3.1 Outdoor Experiments 81
UWB System Setting 81
Experiment Description and Results 83
5.3.2 Preliminary Indoor Experiments 88
Experiment Description and Results 89
5.3.3 Final Indoor Experiments 89
UWB System Setting 91
Experiment Description and Results 91
5.4 Concluding Remarks 96

6 Conclusions **99**

Bibliography **101**

List of Figures

2.1 The Janus protocol. The illustration is a complete account only for node $N1$. Although $N2$ and $N3$ also discover each other during their scans and schedule their own ranging windows, the corresponding portions of the schedule are omitted here for readability. . . . 10
2.2 Modeling Janus: The TWR exchanges of four co-located nodes fail due to a (partial) overlapping, indicated in gray. . . . 14
2.3 Janus nodes used for testing and in commercial deployments. . . . 16
2.4 Latency of first discovery and first ranging for the reactive configuration ($\Lambda = 2$ s for 95% of nodes) in Table 2.1 for different numbers of nodes. These include $N = 10$ (Figure 2.4d), the expected maximum Janus is configured for, and $N = 20$ (Figure 2.4e), showing that performance only marginally degrades despite a system scale twice the expected maximum one. . . . 20
2.5 Latency of first discovery and first ranging for the intermediate ($\Lambda = 15$ s for 95% of nodes) and logging ($\Lambda = 30$ s for 95% of nodes) configurations in Table 2.1 and maximum expected nodes, $N = 10$. . . 21
2.6 Janus ranging reliability estimated by the model (§2.3) and measured in our experiments. . . . 22
2.7 Motion capture setup. . . . 23
2.8 Experimental scenarios and corresponding sample motion traces exported from Optitrack. M and S in the legend stand for Mobile and Stationary node, respectively. . . . 24
2.9 Distribution of the errors between Janus and OptiTrack distances among 6 fixed nodes. The mean, -39 cm, is applied to all Janus measurement to compensate for arbitrary antenna orientations during testing. . . . 24
2.10 Dissecting proximity: LINE. . . . 26
2.11 Arbitrary paths: RANDOM. . . . 27
2.12 Many tags: MULTIPLE. . . . 28
2.13 Current draw for a Janus tag surrounded by 4 others. . . . 29
2.14 Estimated battery lifetime for a Janus tag vs. the percentage of time spent in proximity as a function of the time ratio spent in communication range with one or nine other devices. . . . 30
2.15 Cafeteria: Raw data from one individual. The zoomed-in area shows detail of BLE data. . . . 31
2.16 Small office: Cumulative time of one individual at given distance from others during one workday. . . . 32
2.17 Company-wide: Contacts of 90 individuals over 3 days. . . . 32
2.18 Factory floor: average distance and total time in contact over a 24-hour period for each reported contact dyad. . . . 33

3.1 Janus device management at the AM-PRI camp. Left: An educator fitting the waist bag containing the device on a child, on the first camp day. Right: Devices in waist bags sitting on a storage bench overnight; the inhibitor device is inside the red bag in the center. . . . 41
3.2 Filtering of the spurious measures. Distribution of the measurements over the entire sampling period, either with *Active* or *Inactive* status for AM-PRI. . . . 42

3.3 Measurement splitting and contact aggregation process. The figure shows the measurements collected in the first 20 min of August 8th, 2020, between node 26 and 27 at the AM-PRI camp. The measurements (light colors) are colored according to the division into contact characterized by $\tau_{time} = 90\ s$ and $\tau_{space} = 2\ m$. Each contact is depicted as a horizontal bar from its beginning to its end, where the height of the bar represents the median distance. 43
3.4 Summer camp contacts and contagion risk. The figure reports, for each summer camp, the corresponding contacts classified according to their risk of contagion as a function of the duration of exposure and proximity, following the risk categories in Table 3.5. The values in parentheses denote the percentage of time spent in a contact with the corresponding risk category. 46
3.5 Unique contacts and risk levels. Contacts from AM-PRI, aggregated into a single point (one per device pair) according to different criteria. 47
3.6 Social bubble policy and roles. Distribution of risk levels by group and type of interaction for each summer camp. The color bars, which refer to the right-hand scale, report the percentage of time of contact within each risk level. The grey background bars, which refer to the left-hand scale, report the total time of contact for each of the six categories. 50
3.7 Activities and risk levels. The figure shows the distribution of the risk levels by activity, sorted according to a decreasing percentage of high-risk contacts for AM-PRI (Figure 3.7a), DAY-PRI age range 6-11 (Figure 3.7b) and DAY-INT 11-14 (Figure 3.7c). The percentages show the fraction of contact time within each risk level, for each activity. . 51

4.1 Hermes protocol overview. Details of a single discovery/ranging are shown. 56
4.2 Illustration of Hermes in simulation showing two devices. The red bar shows an instant of time T in which one `trainslot` and one scan slot are simultaneously active, leading to discovery. 57
4.3 Concurrent interactions of slots for more than two nodes. Green arrows represent successful message exchanges, while the red arrows depict a failed exchange. 58
4.4 Hermes jitter. λ is the jitter, and is a uniformly distributed random variable from λ_1 to 0, where λ_1 is a negative value. 59
4.5 Simulations with 10 nodes and 2 s epoch length to understand the impact of collisions (Shadowing and Response collisions on the discovery rate as the `train` length increases). 60
4.6 Discovery rates VS. `train` lengths for varying epoch lengths. 61
4.7 Lifetime of Hermes vs `train` length for a 2 s epoch. 62
4.8 Discovery rates VS. Latency for varying jitter. 63
4.9 Photo of the experiment area in the CLOVES [69] testbed. 7 devices are installed on the ceiling and 8 temporary, *fly-nodes* are placed with wood sticks. 66
4.10 Discovery rates VS. `train` length with 3 nodes. 67
4.11 Discovery rates VS. `train` length with 5 nodes. 67
4.12 Discovery rates VS. `train` length with 15 nodes. 68

5.1 CAD map of the indoor testing area. 73
5.2 Photo of test garage highlighting possible sources of interference (1. Parked cars, 2. Concrete pillars, 3. Metal pipes). 73
5.3 Architecture overview of the UWB test system. 74
5.4 Mounting mechanism for UWB anchors on pillars in the garage environment. Different from the figure, the height was approximately 2 m. 75

5.5 UWB anchor on a movable microphone stand at approximately 2 m height. . . . 76
5.6 UWB antenna and V2X antenna mounted on a test vehicle. The antenna offset is shown. Values of 5 to 9 cm were typical during experiments. . . . 77
5.7 Data processing pipeline enabling comparison between V2X latitude/longitude logs and the UWB XY logs. . . . 77
5.8 Super frame of the DRLTS system [25]. . . . 78
5.9 Details of a single two-way ranging slot in the DRLTS application [25]. 78
5.10 Left: narrow test, small (orange), medium (green) and large (blue) areas. Right: wide test, small, medium and large area. In all cases, anchors are placed at the corners with the antenna facing the vertical axis. . . . 79
5.11 Wide test, small area, showing the three sample points (corner, center and side). . . . 79
5.12 Test area 1 showing 6 anchor locations. Orange arrows indicate antenna orientation. . . . 80
5.13 Test area 2 showing 18 anchor locations. Orange arrows indicate antenna orientation. . . . 81
5.14 Google Earth view of the experiment site. The red arrows indicate the UWB anchors positions, while the two red cars with the colored squares around, the starting positions of the two experiments. (during the experiments fewer cars were parked on the terrace). . . . 82
5.15 Example of UWB anchor installed on a lamppost. . . . 82
5.16 Same part of the same experiment, with and without the index offset compensation. For visualization purposes, the tracks are downsampled. . . . 84
5.17 A full trial of the 8-shape experiment. White and red dots are RTK and UWB data, respectively. . . . 85
5.18 Detail with UWB deviation in one 8-shape trial. White and red dots are RTK and UWB data, respectively. . . . 85
5.19 Distance difference in meters between the GPS and UWB traces for the duration of trace 1. The index corresponds to the 10Hz samples of the UWB trace. . . . 86
5.20 A full trial of the small circle experiment. White and red dots are RTK and UWB data, respectively. . . . 87
5.21 Detail with UWB deviation in one small circle trial. White and red dots are RTK and UWB data, respectively. . . . 88
5.22 Distance difference in meters between GPS and UWB for the first circle experiment. . . . 88
5.23 Trace recorded during the full walk experiment. . . . 90
5.24 Old configuration, zoomed view. Two fewer anchors (red crosses) and one anchor pointing in a different direction (blue arrow). . . . 90
5.25 Track recorded during the snake walk experiment. . . . 90
5.26 Single trace of the simple path with V2X in green and UWB in red. . 92
5.27 In green, all the V2X traces. . . . 92
5.28 In red, all the UWB traces. . . . 92
5.29 The difference measurement in meters over time between the V2X trace and the UWB, applying the index that minimizes the average difference. . . . 93
5.30 Single trace of the curve path with V2X in green and UWB in red. . . 93
5.31 In green, all the V2X traces. . . . 94
5.32 In red, all the UWB traces. . . . 94
5.33 The difference in meters between the V2X and UWB over a single trace with the index that minimizes the average difference. . . . 95
5.34 An anomalous area of the UWB data trace. . . . 97

List of Tables

2.1 Janus configurations used in the system evaluation. 18

3.1 Description of the three summer camps investigated in our study. . . 39
3.2 Daily activities at the summer camps, each with a brief description, the location and the duration in minutes for each summer camp that offered the activity. 40
3.3 Statistics of the raw data sets, including the number of measures before and after the pre-processing step. 42
3.4 **Description of the contacts resulting from the aggregation procedure.** For each camp, we report the total number of contacts, the average number of the measurements for each contact, the number of groups and activities in the camp, and the number and percentage of the contacts that are uniquely associated with an activity. For DAY-PRI and DAY-INT, we report both the number of activities, and the number of activities considered for the analysis (in parenthesis). . 44
3.5 Risk levels of contagion defined on the basis of duration of exposure and physical distance. 45
3.6 **Summary of the number and duration of the contacts in the three camps according to the social bubble strategy.** For each camp AM-PRI, DAY-PRI, and DAY-INT, we report for the different bubbles the total time of contact and the number of contacts organized by the role of the participants. 48

4.1 Sample parameter configurations for Hermes. 56
4.2 Current consumption of Hermes protocol. 68
4.3 Preamble Sniffing evaluation: considering discovery rate and current consumption for various duty cycle configurations. 69

5.1 Wide test area, with cars inside rectangle defined by the anchors. Root Mean Square error in meters reported as the difference between ground truth and measured values across all experiments. . . 80
5.2 Narrow test area, with no cars inside rectangle defined by the anchors. Root Mean Square error in meters reported as the difference between ground truth and measured values across all experiments. . 80
5.3 8-Shape experiment result summary. 84
5.4 Circle experiment result summary. 87
5.5 Index offset and error for the indoor Simple Path without Stop. . . . 91
5.6 Index offset and error for the indoor Simple Path with curve. 96
5.7 Index offset and error for the indoor Simple Path without Stop. . . . 96

Chapter 1

Proximity Detection: a Key Application for IoT

Proximity detection is a fundamental building block of several mobile and Internet-of-Things (IoT) applications.

In many contexts, the ability to detect when devices carried by users are close to others deployed in the environment is exploited to *enable interaction*. For instance, *proximity beacons*, made popular by Apple iBeacon [6] and Google Eddystone [38], are typically deployed in designated places of interest, e.g., the exhibits of a museum, providing a simplified form of localization. When proximity to a user device is detected, some situated user interaction is triggered, either application-specific (e.g., visualization of content about the associated exhibit) or via standard means, as envisioned by the Physical Web [101]. *Proximity warning systems*, which typically rely on different technologies [7], build upon similar concepts to improve safety of the workplace and alert workers when they come too close to potential hazards. These can be static, e.g., containers of dangerous material, but also *mobile*, e.g., operating machinery such as forklifts and excavators in construction sites [48].

In several other contexts, proximity detection is instead exploited to *acquire data, enabling its analysis*. Such logging of user proximity events enables analysis and modeling user behavior, e.g., the time spent near an exhibit. Many other applications focus on collecting solely data about proximity among users to study social interaction. These include *proxemics* [40], the study of the space individuals set between themselves and others, and many others where quantifying *social contact* is key, as in studying relationships [77], or the social behavior of people at scale [93]. Similarly, *biologging*, a recent trend in biology, focuses on proximity detection among animals to understand their interactions and behavior [104] [48].

Interestingly, the recent COVID-19 pandemic fueled a demand for systems along both dimensions. The use of personal devices such as smartphones [3] or dedicated "proximity tags" [16] has been recently proposed towards *i) real-time enforcement* of social distancing, e.g., automatically alerting people when inadvertently coming in close contact, and *ii) monitoring and recording* of distance and duration of a social contact, enabling offline analysis, e.g., to trace the spread of infection from a diagnosed individual or inform predictive models. [48]

Before detailing our contribution to proximity detection we take a step back to precisely define proximity detection and to identify the challenges related to this fundamental element of the IoT paradigm.

1.1 *Who* is Around and at *Which Distance*

Proximity detection can be defined as *"The ability of an IoT device to sense the environment in search of similar devices and to know the distance to them."* From this definition, we identify two primary functionalities: identifying the other devices nearby and measuring the distances to them. The first functionality is accomplished by *Continuous Neighbor Discovery*: a device should be *continuously* vigilant to *discover* other devices, its *neighbors*. This operation is commonly achieved using radio technologies by alternating listening and transmitting on the radio channel. As mentioned

previously, this feature alone can be exploited in many scenarios, making it fundamental for present and future IoT applications. In fact, in recent years many continuous neighbor discovery protocols have been developed [67, 27, 100, 53].

The second building block of proximity detection is the capacity of the device to gain knowledge of the distance from its neighbors, once discovered. This feature is called *ranging* and is the fundamental operation for all localization applications, not only those for proximity detection. In the literature several techniques are present: many approaches are based on the Received Signal Strength Indicator (RSSI), a value related to the received power of the signal. This approach can be very practical, with this value easily available on signal reception in most wireless IoT devices. However, the transformation into a distance value requires the definition of an electromagnetic model, with the Friis formula often being too generic, and on the other side ray-tracing models being too specific for the environment. Another well-known approach consist of using Time-Of-Flight (ToF), in which the propagation delay between the transmission of the signal and the reception at the receiver is exploited to compute the distance, given that the speed of the radio waves can be considered equal to the speed of light ($3 * 10^8 m/s$). Even if with this approach it is possible to achieve high accuracy, this way of calculating the distance is strictly related to precision of obtaining the timestamps of transmitted and received signals, often low due to hardware limitations.

In general, there are many ways to achieve both the discovery and the ranging, each providing a building block for proximity detection, and each with particular trade-offs. In the following sections, we describe several of these technologies.

1.2 Technologies for Proximity Detection

Several technologies have been explored for detecting proximity, e.g., including infrared [21], ultrasound [47] and IEEE 802.15.4 [65]. Bluetooth Low Energy (BLE) devices are a common choice, thanks to its pervasiveness on personal devices and a wide range of consumer electronics, low energy consumption, and availability of a ready-to-use, standardized API simplifying development. Applications include proxemics [40], social studies [70, 2], museums [106], and proximity warning systems [7]. Radio-Frequency Identification (RFID) is another popular choice [18, 77]; in particular, passive tags do not require a power supply enabling very cheap, disposable designs.

Focusing on the radio technologies, RFID and BLE have shown potential of proximity detection for the above classes of applications. Nevertheless, they do not *directly* measure distance, which is instead inferred via the RSSI. This approach typically yields only coarse estimates with errors of meters [110] severely limiting the potential applicability. For instance, in museums their application is usually limited to detecting user presence at the room level [106, 65] or identifying "hotspots". The application in safety-critical settings like proximity warning systems is impaired by the presence of false positives and false negatives, induced by the vagaries of wireless communication in complex environments. These significantly affect also the exploitation of proximity detection for social studies, as recently noted in the context of the COVID-19 pandemic [58]. Further, these social studies typically require sub-meter accuracy [8], which exacerbates the problems above [48].

A promising alternative is offered by Ultra-Wide Band (UWB)[1] radios [16, 17, 10], recently at the center of research and market interest after a decade of oblivion, thanks to small, cheap, and energy-savvy new UWB impulse radio chips. These operate on fundamentally different PHY-level principles that enable distance estimates with an error <10 cm, i.e., 1–2 orders of magnitude less than narrowband radios like BLE, significantly enhancing ranging accuracy. UWB localization systems are rapidly gaining traction and, by yielding accurate and timestamped (x, y, z) positions, *indirectly* enable proximity detection. Nevertheless, they require

[1] In the remainder of the thesis we refer to High Rate PHY-UWB (HRP-UWB).

an infrastructure of fixed reference nodes (anchors), implicitly delimiting the area where detection can occur, posing conflicting trades of scale vs. effectiveness vs. cost. While this may be acceptable in some contexts, it severely limits application in many others where it is impossible or impractical to setup an infrastructure. For instance, longitudinal sociological or epidemiological studies *continuously* monitoring the proximity of individuals as they spend their day across various places (e.g., home, school or work, recreation) would become essentially impossible [48].

On the other hand, UWB systems can also be used in an infrastructure-less, "peer-to-peer" modality where nodes range against each other. Unfortunately, this is precisely where roles are reversed, and popular technologies like BLE shine w.r.t. UWB. Indeed, BLE *directly* supports device discovery via its scan and advertisement operations, at the core of all BLE-based proximity detection approaches; further, it does so very efficiently from an energy standpoint. In contrast, not only does UWB lack similar commonplace protocols, it also has significantly higher energy consumption. For instance, the popular DW1000 UWB transceiver consumes $\sim$80 mA in TX and $\sim$120 mA in RX; the BLE chip on the dual-radio module we exploit consumes only $\sim$4 mA and $\sim$6 mA, respectively [48].

1.3 Challenges for Proximity Detection Protocols

As we have seen in the previous sections, proximity detection requires finding who is around and their distance, dividing the problem into two components: continuous neighbor discovery and ranging. Even if at first glance, proximity detection seems a straightforward task, it carries with it several challenges that one must take into account when designing a protocol.

Ranging, for instance, requires a certain *accuracy* in measurement. If the target environment must identify when two people are within 2 m, an error of 30 cm in the calculation is acceptable. However, if instead we must also collect data *if and only if* the two people are at 2 m or less, those 30 cm matter and therefore the high accuracy becomes fundamental. As mentioned previously, different ranging techniques offer different accuracy levels. Moreover, the technologies here play an important role, with UWB as one of the most accurate radios with an error $<$10 cm.

However, accuracy is not everything. In the IoT paradigm, devices are energy-constrained "Things" that can not rely on bulky batteries, due to the their small dimensions. Therefore, *energy consumption* represents a classical thorn in the side of every IoT task, including proximity detection. While the battery lifetime is one requirement of a specific application, it is generally difficult to predict many of the factors that influence consumption. For example, how long a device will be alone, without seeing any neighbor? How many devices on average will it meet per day? Depending on the answers, the radio activity will change from zero ranging if always alone to a continuous ranging if always in the company of others. A concrete example of this is illustrated in [88], where the authors test an IoT smart door lock and the answers to the previous questions change, depending on the scenarios considered. In general, decreasing as much as possible the energy consumption allows a protocol to be more flexible so that it can be used for a wide spectrum of applications. In this regard, UWB is not the best performer. The consumption cited in the previous section demonstrates that designing a low-power proximity detection protocol using UWB is challenging, while would be easier with BLE for instance, thanks to its low-power radio module.

In the examples that we made before, there was always a mention of the application in which the protocol is going to be used. We can distinguish the use cases exploiting proximity detection in two broad classes: *reactive* ones focused on exploiting proximity for run-time interaction and alerts, and *logging* ones focused on acquiring data for subsequent offline analysis and interpretation. These are not mutually exclusive, and may appear together in a single application. For instance, the recent pandemic fueled a market surge of "proximity tags", geared both towards real-time alarms and offline analysis. Similarly, applications in the

healthcare domain may exploit logging functionality to quantify patient-caregiver interactions, but also exploit alarms to protect specific patients from dangerous situations (e.g., an Alzheimer's patient near the ward exit door) [48]. That is the moment in which the concept of *latency* comes into play. The examples before state that, depending on the type of application, the proximity detection must occur within defined time bounds. Therefore, in a reactive application, ideally, the latency should be lower, permitting proximity detection as soon as possible while, for a logging application, the time bound can be more relaxed.

To summarize, *accuracy*, *energy consumption*, and *latency* represent a triad of variables that one must consider in designing a proximity detection protocol for IoT. Clearly, they are often at odds with one another, posing different challenges to the designer. For example, achieving a low latency protocol is fundamental for a reactive application, however this turns into high consumption, with the radio triggered more often, thus draining faster the battery. Also finding the right trade-off between accuracy and lifetime is tricky. If on the one hand, UWB is a technology that allows one to achieve accurate measurements, on the other hand its consumption is high, making BLE preferable at the cost of ranging accuracy.

1.4 Thesis Statement and Structure

This thesis sets out to strike a balance among *accuracy*, *energy consumption* and *latency* in proximity detection. The core of the thesis revolves around the design and implementation of two proximity detection protocols, in which understanding the equilibrium of the aforementioned variables is at the core. The backbone supporting this thesis is UWB technology: the high accuracy guaranteed by its PHY layer, allows "fixing" one of the variables, ranging accuracy, allowing us to concentrate on energy consumption and latency. This choice is further motivated by the spread of UWB modules in new smartphones [103], making this technology available also to the smartphone application developer in the near future and drastically increasing its ubiquity.

Proximity detection protocol development was not the only motivation for this work. The idea of offering a *complete treatment*, in which the protocols are not "simply implemented", but are also examined in real scenarios is also a fundamental target of the thesis. Therefore, this thesis also demonstrates the real potential of proximity detection, offering a thorough analysis that the protocols enabled.

The first step toward the thesis target is embodied in the Janus protocol [48] described in Chapter 2, in which the three dimensions of the problem are explored with a dual-radio architecture. In Janus, the continuous neighbor discovery and the ranging are split, with BLE being use for the former while the ranging is achieved with UWB, guaranteeing accuracy. The described protocol is both accurate and energy efficient, producing a flexible protocol useful in different environments as shown in the chapter. This protocol represented our first complete work into proximity detection and into UWB technology, thus representing a fundamental experience for this thesis.

Importantly, we also had the opportunity to use Janus in a real experiment [60], as depicted in Chapter 3. Here, we show the potential of Janus for monitoring activities of children at indoor and outdoor summer camps held during the COVID-19 pandemic. This study was conducted in collaboration with a research group working on data analysis, allowing us to develop a complete work in which we show several interesting results enabled by Janus. Indeed, thanks to the availability of useful metadata on the children's activities and to the high accuracy provided by Janus, we offer multiple insights on children's interactions, from the definition of contagion risk levels, to the study of the efficacy of social bubbles between groups. This work clearly shows the enormous potential of studies and analysis that a well-developed proximity detection protocol can permit.

Despite the successful use of Janus, we returned to the protocol, noting that the architecture that splits discovery and ranging across two radios introduces fundamental drawbacks: first, it requires careful coordination between the radios, but more importantly, it introduces latency that imposes a limit on the applicability of the protocol for low-latency applications. Therefore, we decided to return to a single-radio architecture, and design a novel protocol, with UWB as the main technology as shown in Chapter 4. With this new protocol, Hermes, the challenge moves from the coordination of the two radios to the design of the UWB transmission and reception scheduling that is reliable despite collisions, energy efficient despite the use of UWB, and still flexible enough to be applied in a variety of settings. Our contribution here is a low-latency protocol that, while it has higher energy consumption than Janus, its advantages make it applicable in a variety of novel applications. In the chapter we provide the results achieved with a custom simulator, a fundamental tool to test different configurations and explore the trade-offs of the protocol. Moreover, with the real implementation of Hermes, we show that we can achieve reliability in line with the simulator and implement different techniques to reduce the consumption.

Finally, Chapter 5 presents a project on localization with UWB technology, a project not strictly in line with the thesis objectives of proximity detection, but interesting in its practical use of UWB. The project was in collaboration with a company[2] and the target was to create a testing infrastructure to assess the quality of a Global Navigation Satellite System (GNSS) Vehicle-To-everything (V2X) solution for indoor environments. In contrast to the rest of the thesis in which novel protocols were developed and evaluated, here we used a commercial UWB localization system, moving the challenge from protocol design to the validation of the systems. The creation of a complete system to be used as the ground truth, the installation of the system in the field and the data comparison to the tested commercial GNSS solution represent the main contributions of this chapter.

The thesis concludes in Chapter 6 with a summary of the contributions brought to the proximity detection, together with a brief outline of future work.

[2] All details of the collaboration, including the collaborator and the system under evaluation, have been anonymized for contractual reasons.

Chapter 2

Janus: Accurate *and* Energy Efficient Proximity Detection Protocol

A fundamental step toward a proximity detection protocol requires addressing the trade-off between ranging accuracy and energy efficiency. Janus reconciles these dimensions in a dual-radio protocol enabling accurate *and* energy-efficient proximity detection, where the energy-savvy BLE is exploited to discover devices and coordinate their distance measurements, acquired via the energy-hungry UWB. A model supports domain experts in configuring Janus for their use cases with predictable performance. The latency, reliability, and accuracy of Janus are evaluated experimentally, including realistic scenarios endowed with the mm-level ground truth provided by a motion capture system. Energy measurements show that Janus achieves weeks to months of autonomous operation, depending on the use case configuration. Finally, several large-scale campaigns exemplify its practical usefulness in real-world contexts.

The work presented in this chapter has been published in [48].

2.1 Introduction

Named after the god with two faces in Roman mythology, Janus exploits a dual-radio approach taking the best of BLE and UWB: the low-power consumption of the former and the accurate distance estimation of the latter.

In Janus, continuous neighbor discovery is performed by the lower-energy BLE radio, while the higher-energy UWB radio is *triggered on-demand* solely when a device is discovered and a distance estimate is required. This yields the same decimeter-level accuracy provided by UWB, but increases lifetime up to months. However, the UWB distance estimates, obtained via two-way ranging [49] exchanges, face the possibility of collisions. Therefore, we double the responsibility of BLE and use it both for discovery and to *coordinate* ranging exchanges, piggybacking their schedules in the BLE advertisements, increasing reliability and ensuring a predictable behavior.

On the other hand, the accuracy of distance estimation and energy-efficiency are not the only metrics of interest determining the practical usefulness of a proximity detection system; the latency of first discovery, ranging update rate, and reliability all play a key role. The *configurability* of the system is another fundamental requirement; domain experts should be able to tailor the system behavior to their specific needs with a small set of parameters whose effect is well understood.

2.1.1 Goals, Methodology, and Contributions

After elaborating on these requirements (§2.1.2) we present the design of Janus (§2.2), reconciling accuracy with energy efficiency by combining the respective strengths of BLE and UWB in a novel dual-radio protocol. Moreover, an analytical model (§2.3) is the cornerstone enabling domain experts to configure Janus by

navigating the tradeoffs between latency, reliability, and scalability. Our design is embodied in the implementation we concisely describe (§2.4) along with several other hardware and software components enabling the practical use of our solution, notably including a custom tag with a slim, badge-like form factor (§2.5).

We characterize the performance of Janus in our system-centric evaluation (§2.6), using different experimental methodologies. First, we exploit tabletop experiments with up to 20 nodes to retain full control of the *exact* moment nodes come within proximity—a ground truth very hard if not impossible to acquire in the wild due to the vagaries of wireless communication. This enables us to report accurate statistics about the latency of first contact and the reliability of discovery and ranging with a varying number of neighbors, therefore also validating our model. To ascertain the accuracy of distance estimation in realistic scenarios where tags are both worn by people and placed in the environment, we perform a second set of experiments in a motion capture facility providing mm-level ground truth. These experiments also provide unique insights into the effects of body shielding and antenna orientation, factors that affect UWB ranging in general and proximity detection in particular, but are rarely ascertained in the literature [68, 76]. Finally, we show energy measurements confirming that Janus achieves up to months of uninterrupted operation, depending on how aggressively the discovery and ranging latency are configured, and on assumptions about the number of neighbors in range.

The system evaluation is complemented by in-field experiences (§2.7), concretely highlighting the functionality and advantages enabled by Janus. We focus on scenarios related to the recent COVID-19 pandemic as these reunite in a single context the paradigmatic use cases above. A discussion (§2.8) summarizes the key findings of our system and in-field experiments, along with the main tradeoffs entailed in configuring Janus for a given application.

Dual-radio off-the-shelf modules are increasingly common, as exemplified by the popular BLE-UWB Decawave DWM1001C [64] we use here. Interestingly, the same trend is emerging on personal devices, with Apple and Samsung spearheading the use of UWB on their smartphones and smartwatches, amplifying the potential impact of the work described here. Nevertheless, in the context of proximity detection, this dual-radio approach is largely novel, with only a few research and commercial systems. We discuss the significant advantages that Janus brings w.r.t. them and related work (§2.8) before offering brief concluding remarks and an outlook on future work (§2.9).

2.1.2 Requirements

Our high-level goal with Janus is to **support both reactive and logging use cases**, which demands fulfilling several requirements concerning both system performance and ease of deployment.

Among the former, achieving an **accurate distance estimation** *directly* quantifying proximity is a defining feature of Janus, enabled by UWB. In proximity warning systems, inaccurate measurements may cause false positives, unnecessarily raising alarms, or false negatives may expose the user to risk. Similar concerns also arise in logging applications in which an error of meters (rather than decimeters) may undermine validity, with COVID-19 contact tracing applications offering a concrete example.

Moreover, proximity detection must occur within **well-defined time bounds** to ensure prompt user alerting or to correctly capture the time spent in proximity. Two quantities are typically important: *i)* the *latency of first detection* Λ, i.e., the interval between when two devices enter in proximity and when they are actually detected as such, and *ii)* the *update interval* U at which the system verifies whether the nodes are still proximate and at what distance. Their relative importance largely depends on the application; further, "faster" is not always "better". For instance, keeping these values small is an obvious concern for applications

that must trigger an action in real time, e.g., proximity warning systems or proximity beacons. However, a highly reactive detection, desirable in dynamic scenarios with fast moving nodes, must be balanced against the energy consumption caused by the frequent communication it induces; acceptable tradeoffs must be determined by domain experts depending on the use case. On the other hand, the requirements of several logging applications are generally more tolerant. For instance, biologging studies often consider a contact only if two nodes remain in proximity for at least one minute [79]. Moreover, the recent Google-Apple exposure Notification (GAEN) [39] framework at the core of several COVID-19 contact tracing smartphone apps detects proximity via a *fixed* period of 4 minutes.

The latter highlights a relevant aspect of BLE-based approaches: the discovery latency and update rate are typically the same ($\Lambda = U$), as the information used for discovery (BLE advertisements) is also used to estimate distance. This is a very reasonable setup also in Janus, used in the evaluation (§2.6.1) to retain generality. Nevertheless, our dual-radio approach allows us to decouple discovery and ranging, providing domain experts with **extra degrees of freedom** in determining Λ and U independently. For instance, in interactive applications reaction is key, motivating aggressive discovery; distance could be monitored at a slower rate. In many logging applications, the opposite configuration may be preferable; Λ can be large, to filter out transient contacts that would only pollute the dataset with a small U providing fine-grained information for relevant ones.

Nevertheless, all the considerations above hinge on the fact that proximity detection is **reliable**; again, our dual-radio approach meets this requirement along with the intertwined perspectives of discovery and ranging. Indeed, reliability in Janus is determined by the probability to successfully discover a device and subsequently estimate its distance. The reliability of the former affects the latter; if a device is not correctly discovered, the system is oblivious of its presence and its distance cannot be estimated. Further, distance estimation has challenges of its own in dynamic scenarios where globally scheduling ranging exchanges is expensive or even impossible.

This bring us to a second set of requirements, less concerned with the *performance* of proximity detection and more with the *deployment and practical use* of Janus in terms of targeted scenarios and ease of applicability.

A key aspect of Janus is that it makes **no assumptions about mobility patterns** for devices, whose proximity can therefore change in unpredictable ways; after all, characterizing these patterns is precisely the goal in many applications. However, this significantly complicates both timely and reliable discovery, as the system must be assumed to be always in flux. Further, we cannot assume that all nodes remain confined in a given area, which provides two additional requirements. First, an **infrastructure-less** approach is required. This is in direct opposition to conventional localization systems [107] that track users only within a limited area where reference nodes (anchors) are deployed. Second, the system should be **open**, i.e., capable of supporting an arbitrarily large set of overall deployed devices, well beyond the comparatively limited number of those *actually* in proximity at any time. This is crucial to simplify management and liberate applications from co-location or even geographical concerns, therefore enabling the use of Janus in large populations of, e.g., workers moving across areas of the same large organization, or even individuals moving freely across a city to meet friends.

A related, important aspect is the number of devices simultaneously in proximity of another, determining the extent to which the system is **scalable**. An estimate about the expected maximum, typically provided by domain experts, is necessary to adapt the operation of the system to the application scenario at hand, avoiding the under- or over-provisioning of resources in meeting all other requirements. On the other hand, this estimate may be difficult to determine precisely; it is therefore important that the *system performance degrades gracefully* when the actual number of devices in proximity is higher than the expected maximum.

Finally, mobility implies that devices, typically carried by users, must be battery-powered; proximity detection must also be **energy-efficient**—the other defining

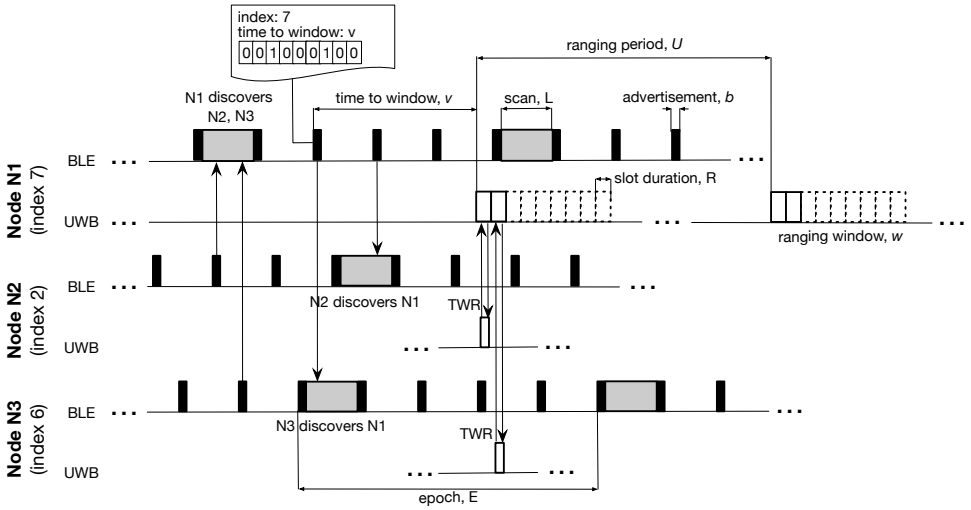


FIGURE 2.1: The Janus protocol. The illustration is a complete account only for node $N1$. Although $N2$ and $N3$ also discover each other during their scans and schedule their own ranging windows, the corresponding portions of the schedule are omitted here for readability.

feature of Janus. The shorter the lifetime, the higher the maintenance overhead and therefore the barrier to adoption. This is exacerbated in use cases where proximity tags are used at scale, e.g., managed by a company to ensure safe working conditions, or in large, city-scale experiments; the cost of frequently recharging hundreds or even thousands of devices cannot be neglected.

Unfortunately, many of the requirements above are often at odds. A high-rate detection yields fine-grained information but reduces reliability, scalability, and battery lifetime. These system-level constraints must themselves be reconciled with the target environment (e.g., construction sites vs. offices vs. wilderness) and other application-specific concerns, sometimes of non-technical nature (e.g., the different duration and distance for "safe contacts" mandated by countries in the COVID-19 pandemic). This demands reconciling system and deployment requirements in a design appropriately balancing them, described next. Nevertheless, supporting domain experts in navigating and optimizing these constraints requires also a **simple configuration yielding predictable behavior**, facilitated in Janus by analytical models (§2.3) whose validity we ascertain experimentally (§2.6.1).

2.2 Janus Protocol Description

2.2.1 Dual-radio Discovery and Ranging

Janus merges BLE-based neighbor discovery and UWB ranging into a single energy-efficient protocol (Figure 2.1) coordinating these two core operations and harmonizing them w.r.t. the requirements above.

2.2.2 Neighbor Discovery

Janus is built atop BLEnd [53], a state-of-the-art neighbor discovery protocol offering configurable, predictable performance. In any BLE-based discovery protocol, a node must transmit *advertisements* to announce itself and *scan* (listen) for those from other nodes, shown as the BLE timeline elements of Figure 2.1. BLEnd provides a periodic schedule for these two operations, therefore enabling *continuous* neighbor discovery. The schedule repeats with period E (epoch, in BLEnd) and

begins with a scan of duration L, followed by advertisements of duration b, determined by the BLE radio, and separated by an interval $L - b$. The relationship between scan duration and advertisement interval in principle guarantees *bidirectional* discovery, i.e., two nodes in range discover each other within a single epoch E. Nevertheless, when multiple nodes execute the schedule in the same neighborhood, collisions may occur that prevent nodes scanning for advertisements from receiving some of them, delaying discovery.

BLEnd takes this crucial aspect into account via a companion optimizer tool that determines the protocol parameters to meet the user requirements in a given setup. Domain experts input the desired latency of first discovery Λ and target probability of discovery P_d, along with the expected maximum number N of nodes in range, which directly affects the collision probability. Based on these requirements, the optimizer relies on an analytical model to determine the values of E and L guaranteeing Λ and P_d, while minimizing energy consumption. The result is an energy-efficient protocol with well-defined discovery guarantees.

2.2.3 Ranging

Among the ranging techniques depicted in the introduction, we select the ToF approach. In particular, Two-way ranging (TWR) is commonly used to estimate distance between two UWB nodes. The simplest variant, single-sided TWR (SS-TWR) is part of the IEEE 802.15.4 standard [49] and requires a 2-packet exchange between an initiator and a responder. The initiator transmits a POLL packet to the responder, which replies with a RESPONSE after a known delay. This packet includes the timestamps marking the reception of POLL and transmission of RESPONSE that, along with the TX/RX timestamps at the initiator, enable it to compute the time-of-flight and estimate distance multiplying by the speed of light in air. Figure 2.1 represents schematically these TWR exchanges on the UWB radio timelines, showing the pairwise messages between two initiators, $N2$ and $N3$, and the same responder, $N1$.

Alternative schemes, e.g., double-sided TWR [49, 73], improve ranging accuracy by reducing the clock drift via additional messages. Instead, we improve SS-TWR by compensating for the estimated clock drift at the initiator based on the carrier frequency offset (CFO) measured during the response packet RX. This technique, recently suggested by Decawave [26, 63], is known [24] to improve the quality of ranging without extra messages.

2.2.4 Coordinating Discovery and Ranging

In Janus both discovery and ranging repeat periodically. Once a node has discovered at least one neighbor, it schedules its own ranging window with a period U, randomized by a small jitter to avoid long-lasting overlaps with those of other nodes. The ranging window contains one slot per discovered neighbor, resulting in a *dynamic* window duration, shown on the UWB timeline for $N1$ in Figure 2.1. In each slot of its ranging window, a node serves as a responder for ranging requests (POLL) initiated by the neighbors. By packing all slots in a single ranging window, a node turns on the radio in listening mode only once per period, reducing the overhead of switching radio states and enabling an efficient use of the deep sleep radio state provided by the UWB transceiver, as described later (§2.4).

As a result, the ranging operation is *asynchronous* w.r.t. that of discovery; UWB ranging windows are scheduled with an arbitrary (and varying) time shift w.r.t. the BLEnd scans and advertisements. Nevertheless, the two procedures are *coordinated* via BLEnd advertisements, which double as a means to inform neighbors about when they should initiate ranging. To this end, a node adds in the advertisement payload *i)* its node index, unique in the neighborhood (see §2.2.7); *ii)* the time v to the beginning of the next ranging window, updated for each advertisement; *iii)* a bitmap indicating the slot allocation for ranging, relying on the node index. Figure 2.1 shows the content of $N1$'s advertisements arriving at $N2$ and

$N3$. Based on this coordination information, both nodes initiate ranging by sending POLL messages to $N1$ in the slots allocated to them in $N1$'s ranging window, obtaining their distance to $N1$. Thanks to the bidirectional discovery of BLEnd, the dual process occurs at $N1$ (not shown in Figure 2.1), providing $N1$ with the distances to its neighbors.

Decoupling discovery and ranging allows Janus to place the overhead of continuous neighbor discovery on the energy-efficient BLEnd protocol, reducing the use of UWB to the bare minimum required for ranging. This is key in scenarios where a node is *not* always in range of some other, and continuous ranging attempts would be wasted. At the same time, the necessary coordination between discovery and ranging is achieved by piggybacking information on the BLE advertisements *that would be sent anyway for discovery*, therefore *i)* at no additional communication and hence energy cost, and *ii)* with the latency and reliability guarantees provided by BLEnd.

Nevertheless, the price to pay for the above is a corresponding decoupling of the *times* at which the discovery and ranging tasks complete, yielding an increased latency of first *ranging* $\Lambda_r > \Lambda$ w.r.t. that of discovery. Indeed, BLEnd guarantees that a node A discovers a newly-appearing node B with latency Λ. However, to perform ranging, B must learn its position in A's schedule via A's advertisements, whose receipt is guaranteed to happen reliably within Λ. The same holds in the opposite direction, yielding a latency of first ranging three times that of discovery, $\Lambda_r = 3\Lambda$, in the worst case. Notably, this does not affect the timing of subsequent ranging estimates, occurring with the desired update rate U; these are typically the crucial ones. Indeed, in many applications the first detection occurs at a distance much larger than the one of interest, e.g., in the case of a person approaching a group of other people. Anyway, in cases where the latency of first ranging must remain below a desired value Λ_r, one can exploit the bound above to set a tighter discovery latency $\Lambda = \frac{1}{3}\Lambda_r$, supporting a faster acquisition of the first ranging. In this case, the inevitable increased energy consumption is nevertheless mitigated by the corresponding optimal configuration output by the BLEnd optimizer.

Finally, slots are allocated for neighbors at the end of each ranging window and de-allocated only after a given number of advertisements are no longer received, indicating the neighbor has moved away.

2.2.5 Time Synchronization

Given the time-slotted coordination of ranging exchanges, time synchronization is crucial to ensure that they complete successfully. Again, we achieve this functionality by relying on advertisements that, according to the BLE standard [49], consist of 3 identical packets sent sequentially on different channels (37→38→39). As each scan occurs on a single channel, changed after each scan, the scanning node receives only *one* of the packets at a *fixed* time offset depending on the position in the sequence. However, since *i)* we verified that the channel sequence is invariant, *ii)* we measured the inter-packet interval in an advertisement, and *iii)* the RX channel is known, the node can easily compute the original time at which the first packet was sent and use it as the reference time to schedule ranging with the sender.

2.2.6 Assigning the Node Index

The ranging window must schedule a slot for each neighbor; depending on the deployment scenario, there may be tens of them. As the schedule must fit into a single BLE advertisement payload (at most $24B$), identifying nodes by their $6B$ address is unfeasible. Instead, we identify nodes with a 1-byte index and advertise bitmaps where a 1 in position x denotes a ranging slot allocated for the node with index x. The slot number is defined as the ordinal number of the 1 in the bitmap. Figure 2.1 shows the first 9 bits of $N1$'s neighbor schedule, specifying that nodes with index 2 and 6 are expected to range in the first and second slot, respectively.

This bitmap must accommodate the maximum expected number of neighbors and minimize conflict among indexes, discussed next. In our implementation (§2.4) the unused portion of the advertisement payload is $13B$ yielding an address space of 104 bits, large enough to satisfy both concerns.

2.2.7 Open system: Resolving Index Conflicts

In large-scale settings, the nodes deployed may be many more than the available node indexes, which therefore cannot be *globally* unique. Still, the protocol described above requires indexes to be *locally* unique, otherwise multiple nodes would share the same slots and their ranging packets would collide. This is a practical concern that arose in some of the real-world experiences described later (§2.7), where hundreds of Janus tags were carried by workers in a large factory plant to monitor their social distancing. Some of these workers travelled between multiple plants on different days, making even the option of a closed-system with site-specific addressing impractical. An open system is instead required, where nodes in range interact opportunistically without global addressing.

We tackled the problem with a simple scheme that dynamically reassigns indexes upon detecting conflicts. At bootstrap, nodes select their index randomly. As advertisements include the sender index and BLE address, receivers can detect conflicts with their index; the node with the lower BLE address picks a different index, avoiding those already in use. In case two non-neighboring nodes with the same index share a neighbor, the latter indicates the conflict in the advertisement payload, forcing both neighbors to select a different index. Finally, the selection of a new index among those available can be greatly improved w.r.t. purely random with negligible computational overhead. In our system, each node caches the bitmaps of all neighbors; the bit-wise `OR` of all these bitmaps and the node's own schedule yields a zero for all unused index values.

2.3 Mathematical Model

2.3.1 Modeling the Success of Proximity Detection

As we mentioned (§2.1), Janus makes no assumption about the mobility patterns of the devices whose proximity it aims to detect. Nodes can come and go of their own volition, continuously changing the neighborhood of each device. Given the dynamicity induced by this highly flexible and practically relevant scenario, globally scheduling the communication required for discovery and ranging is not a viable option. On the other hand, uncoordinated communications may undermine the packet exchanges enabling device discovery and ranging, due to collisions. Their presence is inevitable in an unscheduled setting; however, it is crucial that their impact is *predictable*, presenting domain experts with a simple way to navigate the tradeoffs between reliability and the other performance metrics when configuring Janus for a specific use case.

To this end, we present in the next a model that estimates the probability of successful detection based on the key user configuration parameters. We later show (§2.6) that this model is in very good accordance with experimental results and can therefore be used in practice to inform the configuration of Janus.

2.3.2 Overall Probability of Success

In Janus, the probability P_j of successfully acquiring the distance of a neighboring device depends on the probability P_d that the neighbor is successfully discovered *and* the probability P_r that the subsequent ranging exchange with it completes successfully with a distance estimate. Therefore, $P_j = P_d \times P_r$ holds; the problem then becomes estimating the individual probabilities.

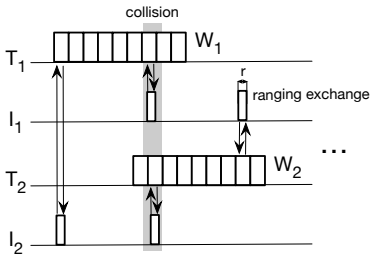


FIGURE 2.2: Modeling Janus: The TWR exchanges of four co-located nodes fail due to a (partial) overlapping, indicated in gray.

Our reliance on BLEnd simplifies matters, as this protocol was expressly designed to ensure predictable performance. The original paper [53] contains an analytical model accurately estimating P_d based on *i)* user requirements: maximum number of nodes N and maximum latency of first discovery Λ, *ii)* BLEnd configuration: epoch duration E and listening interval L, and *iii)* system-level parameters: duration b of a BLE advertisement. Experiments with our Janus implementation (§2.6), based on different firmware and hardware w.r.t. the one used for validation in [53], confirm the accuracy of the BLEnd model in estimating P_d. Therefore, the problem reduces to estimating the probability P_r of successful ranging.

2.3.3 Probability of Successful Ranging

We assumed, without detriment to model accuracy (§2.6), that all N nodes in the system are in range and have already discovered each other. A node performs ranging by scheduling, with period U, a UWB ranging window containing a sequence of $N-1$ slots, each dedicated to one of its neighbors acting as the ranging initiator (§2.2.1). As mentioned, this provides an efficient means to coordinate all ranging exchanges involving the device as a responder. Nevertheless, exactly N ranging windows are contained within any arbitrary interval U. Further, these windows may overlap, as their scheduling is not globally coordinated across devices, potentially generating collisions between ranging exchanges on different pairs of nodes. Our goal is therefore to derive the probability P_r that a given ranging exchange R does *not* overlap with any other.

Figure 2.2 depicts the problem by focusing only on 4 nodes: 2 initiators and 2 responders. We first derive the probability of collision $P_{c,1}$ for a ranging exchange R_1, initiated by I_1 and targeting T_1 within its window W_1, to overlap with a ranging exchange R_2 scheduled within another window W_2. We observe that $P_{c,1} = P_w \times P_s$, where P_w is the probability for R_1 to overlap with the window W_2 itself, and P_s is the conditional probability for R_1 to overlap with a ranging exchange inside W_2. These can be estimated by observing that the probability P_w of a ranging exchange of duration r to (partially) overlap with a window of duration w within the ranging period U can be equivalently modeled as the probability that a randomly placed point falls inside a collision interval of duration $w+r$ within U:

$$P_w = \frac{w+r}{U}$$

Along the same lines, the probability P_s that a ranging exchange of duration r overlaps with another in the interval $w+r$ is the same of a randomly placed point falling inside a $r+r=2r$ interval within $w+r$. The chance of this happening for *any* of the $N-1$ non-overlapping ranging intervals contained in a ranging window is:

$$P_s = \frac{2r(N-1)}{w+r}$$

yielding

$$P_{c,1} = P_w \times P_s = \frac{2r(N-1)}{U}$$

Thus far we considered the probability for a given ranging exchange R_1 to collide with another in *one* out of $N-1$ competing windows. Since the windows are placed randomly and independently within U, the probability for R_1 to *not* collide with *any* of them is equal to the probability that R_1 does not collide with one window in all of $N-1$ independent random trials, or

$$P_r = (1 - P_{c,1})^{N-1} = \left(1 - \frac{2r(N-1)}{U}\right)^{N-1}$$

2.4 Practical Implementation

2.4.1 Implementation Details

Janus runs atop ContikiOS on the popular DWM1001C module by Decawave, combining a Nordic nRF52832 SoC for MCU and BLE and a DW1000 UWB radio. The dual-radio design of Janus complicates its implementation as the activities performed by both radios are time-sensitive and must be coordinated within a single MCU. For the BLE stack, we rely on SoftDevice, a closed-source implementation from Nordic. While convenient for managing BLE activities, it monopolizes all high priority interrupts to guarantee BLE timeliness and does not provide access to its internal schedule, making it impossible to implement a common, synchronous scheduler for both radios. Therefore, we chose to decouple the two subsystems, loosening the UWB timings, which remain under our control, to accommodate unpredictable, overlapping BLE activity.

According to the SoftDevice documentation, its interrupt handlers may occupy the MCU for up to 128 μs, and two consecutive interrupts may occur within as little as 40 μs. For this reason, during the TWR exchange, we programmed the UWB radio to transmit a response 650 μs after RX of the POLL packet, establishing experimentally that this is sufficient to process the POLL and prepare the RESPONSE even if interrupted by BLE activity. We also exploited a DW1000 feature to trigger packet preamble TX before the packet payload is uploaded to the UWB radio, parallelizing activities and giving the MCU an additional 128 μs to fill in and upload the RESPONSE payload.

Providing the BLE advertisements with the interval to the next UWB window presents another challenge. This value is calculated using a dedicated callback (application-priority interrupt) generated by SoftDevice 5.5 ms prior to advertisement transmission. This interrupt has a low priority and therefore may be delayed (e.g., by UWB interrupts), causing errors in the advertised time interval and therefore in device synchronisation. The Contiki system clock, used to schedule Janus protocol activities, is another error source. It is a tick-based clock with a tick of $\sim$1 ms leading to synchronization errors of up to 2 ms due to rounding.

We addressed both these concerns by adding guard times to ranging slots. While the total duration of a ranging exchange is slightly less than 1 ms in our configuration, we experimentally established that time slots of 4 ms are required to ensure that a given ranging exchange does not cross the time slot boundaries. We did consider the tickless RTimer of Contiki, whose higher frequency could achieve smaller slot duration. Unfortunately, however, Contiki does not provide a multiplexing layer for RTimer, meaning that only one activity can be scheduled at a time, a constraint we could not meet with the highly-dynamic Janus protocol.

Finally, we optimized the energy spent by the UWB radio by placing it in deep sleep mode ($\sim$50 nA) whenever possible. This, however, incurs a relatively long delay ($\sim$5.5 ms) to resume operation, as the radio needs time to stabilize its oscillator, motivating the use of contiguous ranging slots to reduce the wake up overhead.

(A) MDEK1001 tag ready to be handed to users.

(B) Our custom tag, complete with badge-like packaging.

FIGURE 2.3: Janus nodes used for testing and in commercial deployments.

Moreover, Janus is designed as a stand-alone, reusable firmware module, whose API sharply separates the core functionality of reporting neighbors and their distance from its use. For instance, this enables developers to define specific notions of proximity or policies for filtering and storing data. Further, the API allows applications to independently (de)activate either radio sub-system and set its configuration.

2.5 From a Prototype to a Full-fledged System

Janus started as a research prototype that we progressively refined to industry-grade level; it is currently integrated in a commercial offer targeting workplace safety. Here we complement the description of Janus, our main focus, with a concise account of other hardware and software components enabling its practical use.

2.5.1 A Versatile Firmware

Janus is designed as a stand-alone, reusable firmware module, whose API sharply separates the core functionality of reporting neighbors and their distance from its use. For instance, this enables developers to define specific notions of proximity or policies for filtering and storing data. Further, the API allows applications to independently (de)activate either radio sub-system and set its configuration.

2.5.2 A Custom Tag

We tested Janus on the Decawave MDEK1001 evaluation kits (Figure 2.3a). These boards are equipped with USB ports and a nice packaging, ideal for development and experimentation. Nevertheless, their hardware is constrained; the integrated, energy-hungry Segger debugger cannot be easily disabled, and LEDs provide the only form of user feedback. These aspects, along with considerations about user comfort when wearing the tag for prolonged periods, motivated the design of a custom tag.

The current version (Figure 2.3b) has a badge form factor ($106 \times 64 \times 13$ mm) and weighs 62 g. Inside the enclosure, the hardware board includes the DWM1001C, a buzzer providing audible and vibration user feedback, 2 LEDs, a multi-functional on/off controller, and an 8 Mbit Flash memory. A rechargeable 950 mAh Lithium-Polymer battery powers the tag. About 2,000 of these tag units are currently deployed at several industrial sites, as part of a commercial exploitation, whose enabling components are described next.

2.5.3 A Complete Solution

Additional elements support large-scale deployments. A *gateway* enables data collection from the tags via the UWB link and upload to the cloud, where data is persistently stored and can be queried and visualized via a *graphical dashboard*. From a hardware standpoint, the gateway is simply a modified tag integrated with an embedded Linux-based system providing Internet connectivity. The fixed gateways also provide *coarse localization* near points of interest (e.g., a coffee machine), as they can implicitly situate proximity detections in their neighborhood. Finally, a simple, effective solution is provided for situations where nodes are not used continuously and are amassed when not in use (e.g., at the concierge during non-working hours). Proximity detection would be both useless and energy consuming. Therefore, when nodes detect a special *inhibitor* node, they automatically enter a stand-by state for a predefined time (e.g., 5 minutes), after which only BLE is activated; normal operation resumes when the inhibitor node is no longer found.

2.6 In-Vitro Experiments

2.6.1 System Evaluation

We evaluate Janus along different performance dimensions. After discussing our reference configurations (§2.6.2) we ascertain the latency of first discovery and ranging, and the reliability of detection (§2.6.3) using a controlled setup enabling precise control of the moment when a node becomes in proximity, and removing the vagaries arising from motion and other aspects impacting wireless communication. We then analyze the ranging accuracy in a real environment, where we compare the estimates acquired by Janus against the high-rate, mm-level ones acquired via a motion capture system (§2.6.4). The representative proximity patterns we experiment with enable us to quantify directly the threats to accuracy posed by antenna orientation and body shielding. Finally, we confirm the energy efficiency of Janus (§2.6.5) with real measurements, showing it achieves an extended lifetime in configurations of practical interest.

2.6.2 Overall System Configuration and Application Scenarios

We illustrate the key configuration parameters used throughout the evaluation.

Application Requirements and Janus Configuration

Janus is designed to cater for both reactive and logging applications (§1.2). These are differentiated mainly by their *time bounds* concerned with the latency of first

TABLE 2.1: Janus configurations used in the system evaluation.

	Reactive	*Intermediate*	*Logging*
Target properties			
Latency of first discovery, Λ	2 s	15 s	30 s
Ranging update interval, U			
Probability of successful discovery within Λ, P_d	95%		
Expected maximum number of nodes, N	10		
Optimal BLEnd configuration			
Epoch duration, E	1 s	15 s	30 s
Scan duration, L	77 ms	341 ms	353 ms

discovery Λ and ranging update rate U, with reactive applications obviously exploiting tighter time bounds.

The Janus configurations we use (Table 2.1) arise directly from our in-field experience with several real-world applications, including those reported later (§2.7.1) where latency values were determined in conjunction with (or sometimes even mandated by) the application stakeholders. The reactive configuration shown is the one in use by companies exploiting our custom tags (§2.5) in a factory environment to alert people when they are too close to each other (social distancing) or to specific objects in the environment (proximity warning). The logging configuration was similarly used in our in-field experience with tracing social contacts. The intermediate configuration gives us the opportunity to explore a balance between these two extremes. Alternative applications, such as biologging [104], commonly exploit even longer latencies, further relaxing performance requirements. At the other extreme, reactive applications with significantly shorter latencies incur very high energy consumption and therefore are not the target of our work, as further discussed later (§2.8).

For all these configurations, we set $\Lambda = U$, as this choice is simple, general, and has proven practically useful in the experiences above. Alternative choices decoupling the two values towards specific application or system requirements are nonetheless possible. The value of Λ is also part of the input to the BLEnd optimizer, along with the target probability of discovery P_d, and the expected maximum of neighboring nodes, N. The corresponding values for the BLEnd epoch and scan interval output by its optimizer are shown in Table 2.1.

Radio Configuration

The UWB radio uses channel 5, a pulse repetition frequency (PRF) of 64 MHz, a preamble length of 128, and the fastest data rate available of 6.8 Mbps. The BLE radio uses the maximum TX power of 4 dBm and the data rate of 1 Mbps. Alternative tradeoffs between range and energy-efficiency can be set via the firmware API (§2.4), as we further discuss when reporting on our real-world experiences with Janus (§2.7.1).

2.6.3 Latency and Reliability

We use a controlled setting to verify whether the target latency of first discovery Λ in our configurations (Table 2.1) is met by our implementation, and quantify the extra delay incurred by the first ranging. We show that our model of ranging reliability (§2.3) is in very good agreement with our experiments. Moreover, we show that both latency and reliability are affected only marginally even when the number of nodes in range is twice the maximum one Janus is configured for.

Experimental Setup

Determining the latency of first discovery requires the ability to control precisely the instant at which a node enters into range. Unfortunately, due to the vagaries

of wireless communication, acquiring this ground truth is very hard, if not impossible, to do in a real environment. Similarly, reliability could be impaired also by application-dependent external factors. For these reasons, we exploit a setup in which all Janus devices are statically placed on a table, well within their communication range and in line of sight. All nodes are connected to a computer via USB, providing power and the ability to easily collect data logs. This setup allows us to collect more information than by relying solely on the on-board memory, run hours-long tests unattended, and accurately timestamp relevant events by using the single time reference provided by the computer. The latter is crucial for measuring the latency in discovering a node joining the system, an event we can easily and accurately emulate in this setup by simply turning Janus on and off on a tag at a designated, timestamped time. This setup also removes elements like mobility and body shielding, which we return to in §2.6.4.

We explore all configurations in Table 2.1, and study the performance of Janus when the number of nodes is lower than the maximum expected one of $N = 10$, but also when it is *twice* this scale. The latter experiments allow us to ascertain the performance of Janus *beyond* the worst-case it is configured for, where latency and reliability guarantees no longer hold, therefore investigating an important dimension of scalability.

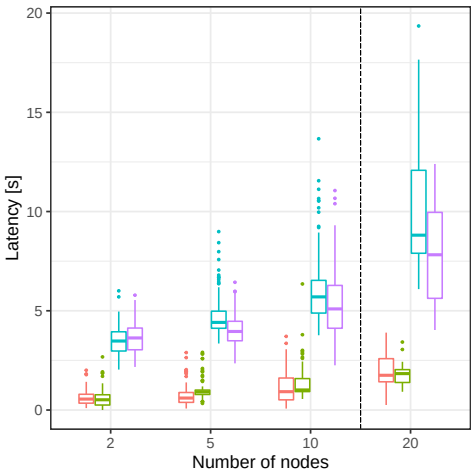
Finally, to avoid biasing the communication schedules of discovery and ranging performed by the two radios, we randomize the node start time within the ranging update interval, U.

Latency

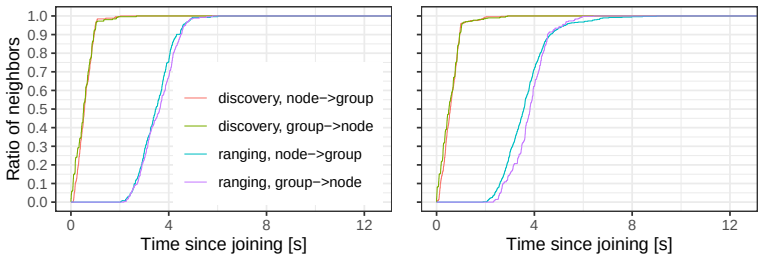
We want to understand the time required by a node approaching a group of other nodes to discover and range with them (node $\rightarrow$ group) and vice versa (group $\rightarrow$ node), i.e., the time it takes the others to discover and range with the approaching node. We mimic this dynamic scenario by modifying the behavior of one of the nodes in our setup to enable and disable Janus periodically, effectively joining and leaving the group formed by the remaining nodes. We ensure that the time with Janus active is long enough for the joining node to discover and range with all its neighbors, and vice versa. We also ensure that the inactive gaps, emulating leaving the group, are long enough to allow all other nodes to remove the departing one from their neighbor tables and ranging schedules. We run the tests long enough to allow at least 100 join events to happen; this is a relatively large number, yielding reliable averages of the relevant metrics, yet manageable in terms of experiment time, considering the several configurations examined. In post-processing, we calculate the difference between the Janus activation time of the "joining" node and the subsequent discovery and ranging events of all nodes.

We run tests for all the configurations in Table 2.1 and show results in Figure 2.4–2.5. The boxplots (Figure 2.4a, 2.5a) show statistics for the time it takes the joining node to discover and range with *all* its neighbors, as well as the opposite, the time it takes all the neighbors to discover and range with the joining node. We also show how the ratio of the discovered/ranged neighbors grows over time since joining (Figure 2.4b–2.4e, Figure 2.5b, 2.5c).

We see that discovery *always* meets, and in many cases exceeds, the target probability (95%) within the target latency ($\Lambda = \{2\ s, 15\ s, 30\ s\}$) for each configuration. This confirms published results [53], of interest given the significant differences in the BLE platforms employed. This result holds both from the perspective of a node discovering the others and vice versa, with the former being only marginally slower. Moreover, Figure 2.4 shows that, as the number of nodes increases, the average discovery time gradually approaches the target discovery latency due to collisions among BLE advertisements, whose effect is nonetheless accounted for in the BLEnd configuration (Table 2.1); the discovery latency meets the target 2 s latency for 95% of the nodes, and never performs worse. Interestingly, this holds even when the system operates outside the guaranteed worst-case, with twice the

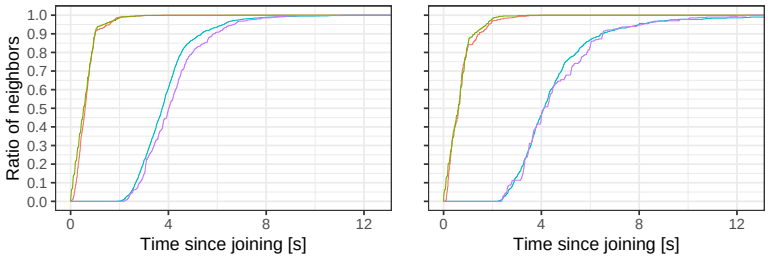


(A) Time to discover/range with all nodes.



(B) 2 nodes.

(C) 5 nodes.



(D) 10 nodes, expected maximum.

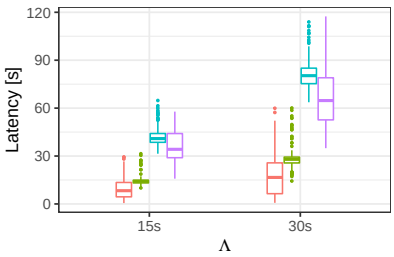
(E) 20 nodes, twice the maximum scale Janus is configured for.

FIGURE 2.4: Latency of first discovery and first ranging for the reactive configuration ($\Lambda = 2$ s for 95% of nodes) in Table 2.1 for different numbers of nodes. These include $N = 10$ (Figure 2.4d), the expected maximum Janus is configured for, and $N = 20$ (Figure 2.4e), showing that performance only marginally degrades despite a system scale twice the expected maximum one.

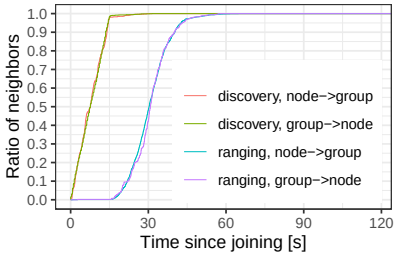
nodes w.r.t. the expected maximum scale it is configured for, confirming the scalability of Janus.

As for ranging exchanges, we recall from the Janus protocol description (§2.2.1) that they are coordinated via BLE advertisements; bidirectional discovery is required before ranging can be scheduled. The charts confirm that, in the best case, the *first* ranging involving a joining node begins after a single U interval, during which coordination occurs and the ranging window is scheduled. However, the charts confirm also that, in the worst case, this first ranging estimate may become available at some nodes only within 3Λ. On the other hand, the CDFs show that $\sim$50% of the rangings occur within 2Λ. Therefore, assuming a random distribution of the ranging success, at least one node in each pair obtains the distance of the other within 2Λ, on average.

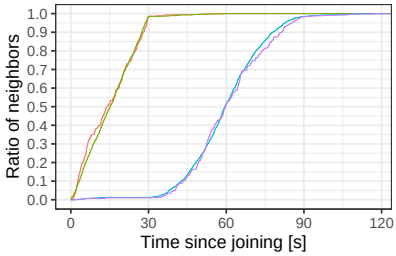
Finally, we observe that, as the number of nodes increases, the average latency



(A) Time to discover/range with all nodes.



(B) $\Lambda = 15$s



(C) $\Lambda = 30$s

FIGURE 2.5: Latency of first discovery and first ranging for the intermediate ($\Lambda = 15$ s for 95% of nodes) and logging ($\Lambda = 30$ s for 95% of nodes) configurations in Table 2.1 and maximum expected nodes, $N = 10$.

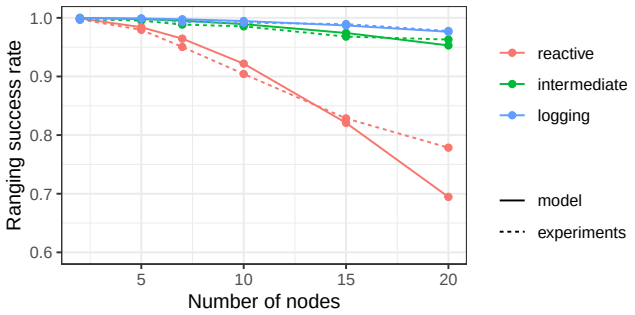
to range also increases due to failed exchanges induced by collisions among the ranging windows. This is expected and quantified by the model (§2.3), whose validity is show next.

Reliability

The controlled setup also enables to accurately estimate the probability that ranging exchanges successfully complete. We elused the same three configurations considered thus far, and analyze the impact of different numbers of nodes, both smaller and larger than the maximum expected, $N = 10$. We evaluated the steady-state performance by allowing enough time for nodes to discover each other before collecting metrics. We ran each test for a duration of at least $120 \times U$, which effectively enabled to measure reliability with a resolution <1%, reasonable in our context; we further report average over 4 repetitions, which nonetheless exhibited only minor variations.

The experimental results (Figure 2.6) show that the intermediate and logging configurations achieve a probability of successful ranging $P_r \geq 95\%$ even with $N = 20$ nodes, i.e., twice the maximum Janus is configured for (Table 2.1). In these cases, the relatively long ranging update interval U can accommodate all 20×19 ranging exchanges with only few collisions. In contrast, the shorter U interval set for the reactive configuration increases the likelihood of collisions, therefore decreasing the success rate, which nevertheless yields $P_r \geq 90\%$ up to the expected maximum of 10 nodes.

The key point, however, is that the impact of collisions can be *predicted* and therefore, if and when needed, compensated for with an alternative configuration that can be explored analytically. Indeed, Figure 2.6 also plots the corresponding values of P_r estimated by our model (§2.3). We can observe that experimental and analytical data are in very good agreement, with the difference generally within 1%. Interestingly, the only exception to this is for the more challenging reactive case and with the highest number of 20 nodes tested, where the model slightly deviates ($\sim 8\%$) from experiments by *under*estimating the probability of success. This is a result of the conservative model assumption that *any* overlap between



	Ranging success rate [%]					
	$U = 2$s		$U = 15$s		$U = 30$s	
N	exp.	model	exp.	model	exp.	model
2	99.75	99.90	99.77	99.99	99.66	99.99
5	97.95	98.41	99.53	99.79	99.88	99.89
7	95.04	96.45	98.85	99.52	99.27	99.76
10	90.43	92.19	98.55	98.93	99.17	99.46
15	82.85	82.09	96.80	97.42	98.94	98.70
20	77.85	69.46	96.30	95.29	97.76	97.62

FIGURE 2.6: Janus ranging reliability estimated by the model (§2.3) and measured in our experiments.

two ranging exchanges results in a failure. In the implementation, however: *i)* actual transmission does not occupy the whole 1-ms ranging exchange and POLL or RESPONSE packets belonging to one exchange may "sneak" in between those of another without causing a collision, and *ii)* the UWB radio can often decode one of the overlapping transmissions [99]. These effects, all beneficial, are more likely to cause a difference in scenarios where the likelihood of collisions is very high, as in the case where we observe the discrepancy between model and experiments.

2.6.4 Ranging Accuracy

We now describe the evaluation of the accuracy of the distance measurements obtained by Janus by comparing them against those obtained via an OptiTrack motion capture facility recording the ground-truth (mm-level) locations of Janus tags over time. This represents also the opportunity to investigate the effects of antenna orientation and body shielding, which degrade the decimeter-level accuracy commonly associated with UWB.

Experimental Setup

In all experiments, we used the reactive configuration in Table 2.1. The tags were continuously in BLE communication range, all discovered before the start of the experiment.

Tags and motion capture. The motion capture facility (Figure 2.7a) offers a 10×8 m^2 area covered by 14 cameras connected via a dedicated Ethernet network. In this space, the OptiTrack Motive system provides millimeter-level accuracy tracking for objects *marked* with appropriate ball-shaped markers.

We performed experiments with stationary and mobile tags, both encompassed by our requirements (§2.1). Stationary tags were mounted at chest height on plastic or wooden poles, and equipped with markers (Figure 2.7b). Mobile tags were instead worn on lanyards around the neck of researchers in our group (Figure 2.7c). Attaching the markers to the tags proved difficult; the OptiTrack system frequently lost track of them, resulting in unacceptable gaps in the measurement trace used as ground truth. Therefore, we attached two markers to the person's shoulders. This

(A) The motion capture area showing six stationary Janus nodes used for compensation calculation and in MULTIPLE.

(B) A single stationary node with motion capture markers.

(C) Tester with markers on the shoulders and a node on a lanyard.

(D) Side view showing the marked Janus node for offset calculation.

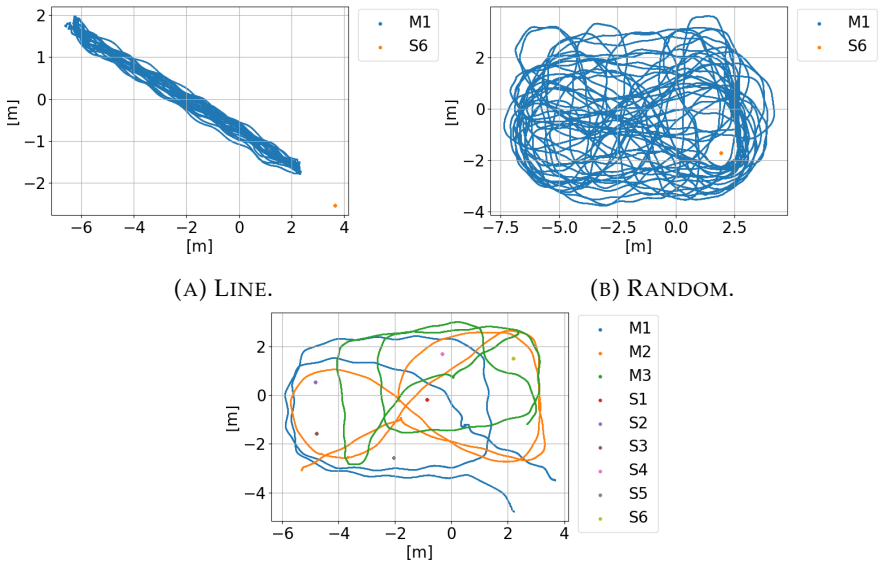
FIGURE 2.7: Motion capture setup.

improved tracking, but introduced an offset between the position automatically identified by OptiTrack (the center of the segment connecting the markers) and the actual position of the tag. Therefore, we used OptiTrack itself to accurately measure the offset for each tester (Figure 2.7d) and exploited an OptiTrack software option to output each position with a fixed, rigid offset w.r.t. the markers, accounting for orientation. This automatically-compensated position trace is the one used in all experiments.

Mobility scenarios. We structure our experiments around three scenarios of increasing complexity, each illustrated in Figure 2.8 with one of the actual Optitrack traces. In LINE, our first scenario, a tag is worn by a user moving back and forth on a line, approaching and retreating from a single stationary node. This very simple setup offers a baseline for analyzing the quality of proximity detection and to ascertain the impact of changes in antenna orientation and body shielding, both occurring when the user walks away from the stationary node. The second scenario, RANDOM, analyzes these effects at scale with a user moving along random, unconstrained paths around the fixed node. Finally, our last scenario, MULTIPLE, further increases complexity with 3 users moving around 6 stationary nodes in patterns mimicking real-world situations, described later.

Measuring distances. In all scenarios above, the (x, y, z) coordinates for all marked objects, captured at 120 Hz, are used to calculate over time the pairwise, 3D distance between all pairs of tags. The pairwise distances measured on the Janus tags were saved in RAM. The tags were time-synchronized to the computer running the OptiTrack software, enabling correlation between the timestamped measurements on the tags and those from the motion capture system.

We calculate the error of each Janus measurement between two tags w.r.t. the OptiTrack ground truth as follows. Based on the Janus timestamp, we identify the corresponding OptiTrack positions for the same tags at the timestamp closest to the Janus one, usually within ± 4 ms. On rare occasions, e.g., when the markers on either or both tags are occluded, the timestamp difference may be higher; if the value



(A) LINE. (B) RANDOM.

(C) MULTIPLE.

FIGURE 2.8: Experimental scenarios and corresponding sample motion traces exported from Optitrack. M and S in the legend stand for Mobile and Stationary node, respectively.

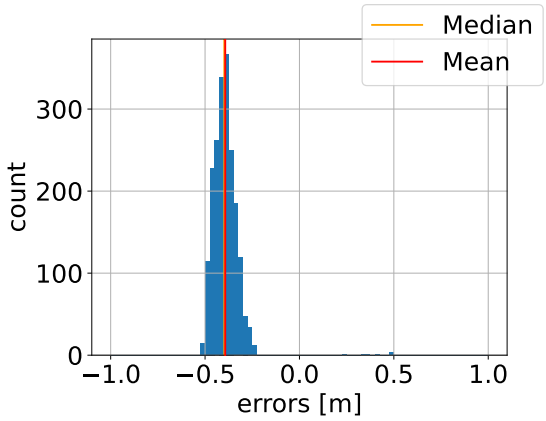


FIGURE 2.9: Distribution of the errors between Janus and OptiTrack distances among 6 fixed nodes. The mean, -39 cm, is applied to all Janus measurement to compensate for arbitrary antenna orientations during testing.

exceeds ± 16 ms, we do not report it for the corresponding Janus measurement. In MULTIPLE, our most complex scenario, only 71 out of 15833 Janus measurements (0,44%) are dropped, and even fewer in the other, simpler scenarios.

The UWB ranging measurements themselves pose a challenge. Indeed, in the absence of specific assumptions, the tags in proximity will have an arbitrary relative antenna orientation, an aspect known to significantly affect ranging accuracy [10]. As such, the usual calibration methods with both UWB tags nicely facing each other are inappropriate. This holds in real-world settings and therefore also in our experiments, which try to reproduce common patterns in a realistic way. To account for this, and provide a reliable accuracy baseline for our experiments, we proceed as follows. We performed a single, 2-minute experiment collecting all pairwise distances among 6 stationary nodes spread unevenly throughout the area, pointing in random directions. This yields the distribution in Figure 2.9, whose average error w.r.t. ground truth is -39 cm. We directly apply this value in

post-processing to all Janus UWB measurements, effectively re-centering (on average) their distributions around this baseline. Alternative methods may be used depending on the application, e.g., when the majority of detections occur along a predefined path with the antennas facing each other, as in our LINE experiment.

Results

Before reporting the ranging performance in our experiments, we note that they enable us to evaluate the overall Janus reliability $P_j = P_d \times P_r$ in a realistic environment, yet with all tags always in BLE and UWB range, similar to the tabletop experiments (§2.6.3). In that case, a direct, wired connection towards all nodes enabled us to report the ratio between successful and scheduled rangings; instead, here we must rely solely on the on-board storage, limiting the detail of logged information. Therefore, we report the ratio between the measurements recorded by Janus w.r.t. the *maximum possible* that could take place in the same time interval.

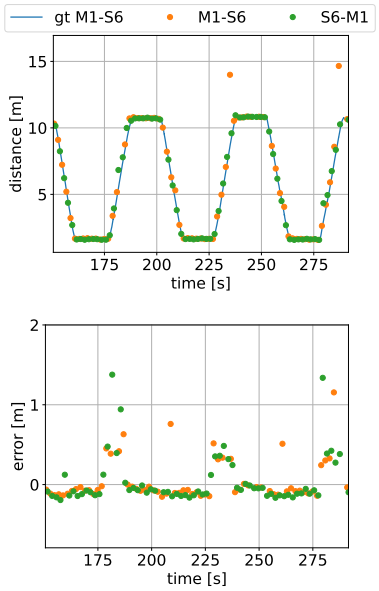
In the simpler experiments, LINE and RANDOM, we observe $P_j = 97.5\%$ and $P_j = 97.1\%$, respectively. Interestingly, the lost measurements are unlikely to be attributable to collisions, given that only 2 nodes are present in these experiments. Instead, they are due to the fact that the radio environment is more complex than in the previous tabletop setting, especially due to body shielding. This affects not only the ranging accuracy, as discussed next, but also the overall reliability, either *i)* directly due to a failed ranging exchange, or *ii)* indirectly due to lost BLE advertisements, which prevent the correct scheduling of ranging windows. On the other hand, in the more complex, 9-node MULTIPLE, we observe $P_j = 88.2\%$ that, once corrected for the minor losses above, is in line with the product of the ranging reliability $P_r = 93\%$ estimated by our model and the target discovery probability $P_d = 95\%$.

Dissecting proximity: LINE. We began our exploration of the ranging performance with a simple, controlled scenario with a tester walking back and forth. We placed one stationary node (S6) near the edge of the area. The tester (M1) began ∼10 m away and walked toward the fixed node, pausing at a distance of ∼1.5 m for ∼15 s. He then turned around, walked back to the starting position, turned back to face the node, paused for 15 s, then repeated the process, for a total of 10 minutes (Figure 2.8a). The experiment was repeated 3 times.

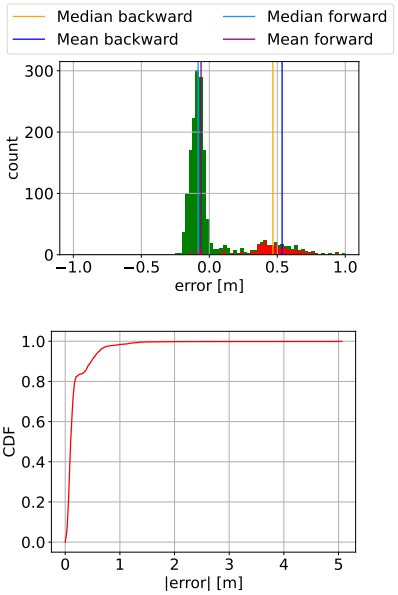
The top of Figure 2.10a shows part of the trace, with large green and orange dots denoting Janus measurements (e.g., M1–S6 indicates those saved at mobile node M1) and blue points denoting the distances computed from OptiTrack positions, our ground truth; these occur at higher frequency (120 Hz) and appear as a continuous line.

We immediately observe that Janus is quite accurate, with most measurements coinciding with ground truth. Notably, there are a few exceptions; in this brief trace, we see two outliers at >11 m. Interestingly, they both occur when the tester is turning around, likely due to an unfortunate combination of antenna orientation and body shielding. These spurious, unreasonable measurements are easy to identify and remove, either online or in post-processing. Indeed, in the data reported here, we filter all values >11 m, as these were unreasonable for the area being studied. Similar arguments can be made on a per deployment basis, removing such impossible measurements either in post-processing or at run-time. In total, in LINE, we removed 2.7% of the Janus measurements exceeding 11 m, a relatively high percentage due to the fact that the tester was often near 11 m. In our later experiments, we removed fewer, 1.5% and 0.4%, as the scenarios measured fewer distances near the 11 m mark.

We next consider the absolute error $\epsilon = d_j - d_g$ between the measured distance d_j and the ground truth one d_g, shown for the same trace portion in the bottom of Figure 2.10a; the error of the two spurious measurements above are removed, as they are beyond the scale. We note two trends. When the tester has his back to the node, walking away, ϵ is larger and positive, i.e., Janus measurements overestimate w.r.t. ground truth. This is explained by the presence of the body between the two



(A) A snippet showing measured distances (top) and error (bottom).



(B) Histogram and CDF of errors for all LINE experiments.

FIGURE 2.10: Dissecting proximity: LINE.

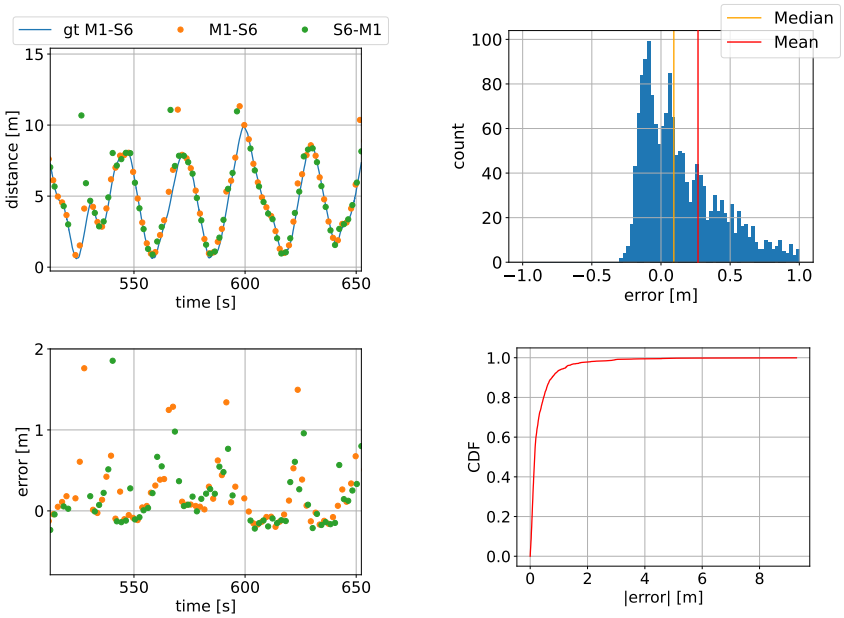
nodes, which slows down the UWB signals, increasing the estimated distances. Instead, when the tester is facing the fixed node, ϵ is slightly negative, between -20 cm and -5 cm, underestimating w.r.t. ground truth. This is a consequence of the procedure we described to establish the accuracy baseline by accounting for antenna orientation (§2.6.4), which accounts for the case (common to all the experiments) of arbitrary orientation, but leads to underestimation in this case with tags in the ideal position.

These two trends are clearly shown in the histogram[1] of Figure 2.10b in which all points were manually annotated to reflect orientation. As the majority of points are recorded with the tester facing the node, shown in green, the overall mean (3.6 cm) and median (-7.2 cm) values of ϵ are shifted negatively, while the errors clustered around 50 cm, shown in red, are those occurring when the tester is walking away. The CDF, which reports only the magnitude $|\epsilon|$ of the error, shows very good results; despite the long tail, the 75^{th} percentile is 16 cm and the 90^{th} is 54 cm. The former is in line with prior validations of UWB ranging, and the later dilution of error is a clear result of body shielding. Indeed, as shown in [68], the body shielding can lead to notable errors. The slight knee in the CDF is due to the high fraction of points with the nodes facing one another.

Arbitrary paths: RANDOM. The previous experiment is intentionally simple to identify threats to ranging performance, specifically the combination of antenna orientation and body shielding. In the next experiment, we analyzed the effect at scale of these perturbations. We still employed only one stationary and one mobile tag, but this time the tester carrying the latter moved along an unconstrained, randomly chosen path throughout the monitored space. The test lasted 15 minutes and was repeated 2 times, covering a large portion of the area (Figure 2.8b) and effectively exploring at once several combinations of distance, relative antenna orientation, and body occlusion between the two tags. As such, it can be regarded as capturing the average performance one can expect in the absence of specific assumptions about these aspects.

The trace snippet (Figure 2.11a) shows a pattern similar to the one in LINE (Figure 2.10a). Overall, the Janus measurements closely follow the ground truth, but larger errors occur when moving away from the fixed node, with a few outliers

[1] For readability, all histograms are cropped to ± 1 m, with the CDFs used to report percentiles.



(A) A snippet showing measured distances (top) and error (bottom).

(B) Histogram and CDF of errors for for RANDOM experiments.

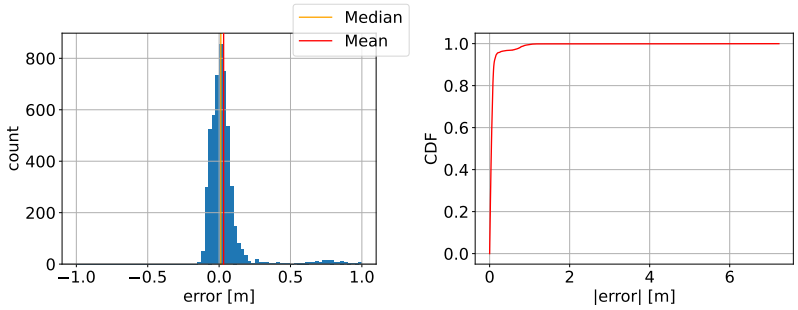
FIGURE 2.11: Arbitrary paths: RANDOM.

often corresponding to changes in direction. Nevertheless, by comparing the histograms of this RANDOM experiment (Figure 2.11b) vs. the LINE one (Figure 2.10b) we see that the errors are significantly more spread in this case due to the arbitrary movement and orientation vs. the rigid and limited ones in LINE. The CDF shows a similar, rapidly increasing distribution, with the 50^{th} percentile at 15 cm and the 90^{th} at 75 cm. This slight degradation is expected, again due to the larger proportion of measurements influenced by body shielding in combination with the random antenna orientations.

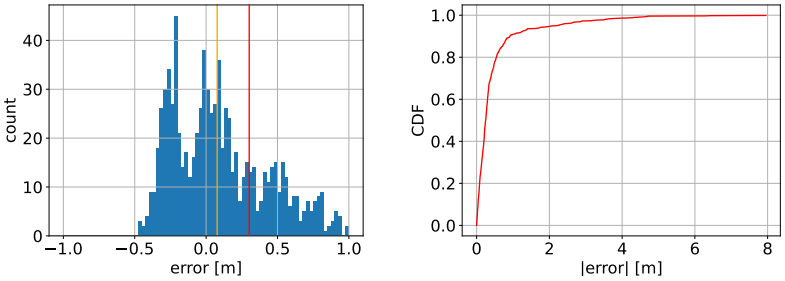
Many tags: MULTIPLE. The final experiment involved 9 tags. It both enabled to assess the reliability and performance of Janus when several detections occur and is representative of multiple real-world situations. We stationed 6 fixed nodes throughout the area, at the same locations used in Figure 2.9; these nodes may represent office workers sitting at desks or designated objects of interest. The remaining 3 tags are carried by testers; they started near the corners of the room, moved over to cluster around node S1, and paused at ∼1.5 m from one another for 15 s, as if to have a conversation. Finally, they departed in an unconstrained way (Figure 2.8c). The entire experiment lasted about 1 minute and was repeated 5 times.

In this complex scenario, the trace snippets are not very informative. Instead, we focus on the cumulative errors, by analyzing different combinations of fixed and mobile tags. We first focus on the subset of measurements among fixed nodes only (Figure 2.12a) and observe that the results are well aligned with those in Figure 2.9, except for the appearance of a few, significantly overestimated distances. This is due to the body shielding randomly induced by the 3 testers, causing an effect akin to that observed in LINE. As seen in the CDF, the data is exceptionally accurate, with the 90^{th} percentile showing only 11 cm.

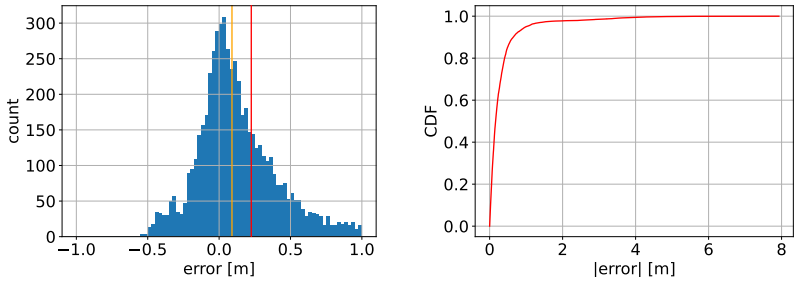
At the other extreme is the subset of measurements entirely among mobile tags, representative of social distancing among individuals, or proximity warning w.r.t. moving machinery. In this case (Figure 2.12b) the distribution of errors is wider, akin to that in RANDOM but more irregular, as a consequence of the relative movements among tags, mixed with periods in which they are stopped. Further, in many cases the testers are back to back, with the two bodies shielding the UWB signals and increasing measurement errors. Although the accuracy drops in these



(A) Fixed-Fixed.



(B) Mobile-Mobile.



(C) Mobile-Fixed.

FIGURE 2.12: Many tags: MULTIPLE.

conditions, the 50^{th} percentile is 25 cm, the 75^{th} is 46 cm, and the 90^{th} remains at 91 cm, i.e., sub-meter.

Finally, we consider the subset of measurements involving one fixed tag and one mobile tag (Figure 2.12c) representative, e.g., of scenarios where a proximity alarm could be raised upon approach to a given location. Unsurprisingly, the results are a mix of the two previous ones. Distances are slightly overestimated, with a better median and mean error than in the mobile-only case. Similarly, the CDF at the 50^{th} percentile is 16 cm, while the 75^{th} and 90^{th} are 35 cm and 64 cm, respectively. This slightly better performance emerges because one of the tags is fixed, which both induces a less irregular error pattern and reduces the impact of body shielding.

Overall, this last, realistic experiment, validated against ground truth, confirms that Janus delivers accurate sub-meter estimates. We now turn our attention to its other defining feature, energy efficiency.

2.6.5 Energy Consumption

We investigate energy consumption, and therefore the lifetime achieved by Janus, by acquiring current draw measurements with a Keithley SourceMeter 2450. We used the custom tags (§2.5), as they do not suffer from the power limitations of the

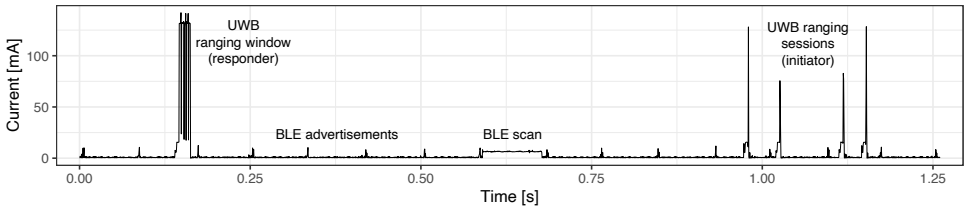


FIGURE 2.13: Current draw for a Janus tag surrounded by 4 others.

MDEK1001 devices. We remind the reader that the battery's capacity of the custom tag was 950 mAh.

This also provides us the opportunity to peek at the actual behavior of a Janus tag executing the protocol schedule (§2.2.1) for which we show a trace segment (Figure 2.13) acquired at a sampling rate of ∼2 kHz. The protocol phases are clearly distinguishable: BLE advertisements (low, periodic peaks), UWB ranging sessions (high, aperiodic peaks), BLE scan (low interval) and UWB ranging window (high, shorter interval). The trace concretely shows the significantly lower consumption of the BLE radio w.r.t. the UWB one, as well as their interplay.

We compare the three representative configurations in Table 2.1. To estimate battery lifetime we observe that real-world scenarios are a mix of periods where the user is alone and others in contact; however, the exact proportions of the mix are typically application-dependent and unknown a priori. To account for this, we explored the configurations in three scenarios: when a tag is alone and when in contact with exactly 1 and 9 others. The first scenario serves as an upper bound for lifetime; combined, the three scenarios enable us to investigate different proportions of alone vs. in-proximity times, spanning several operational conditions at once.

Figure 2.14 shows the results, based on averages over several 15-minute traces. When a tag is alone, only BLE is active, performing neighbor discovery via the BLEnd protocol. As this contribution is invariant w.r.t. the number of neighbors present, this enables us to characterize the energy consumption due only to BLE in the various configurations: the average current draw ranges from 1.1 mA (reactive) to 0.61 mA (logging), yielding a lifetime from 36 to 65 days. When neighbors are present, the triggering of UWB increases consumption, with a significantly different impact depending on the use cases. With our logging configuration, the current draw increases only to 0.65 mA for 1 neighbor and 0.81 mA for 9 neighbors; instead, the reactive configuration increases current draw to 1.58 mA and 4.07 mA, respectively. These trends are reflected in the slopes of lifetime curves (Figure 2.14), which can be easily estimated based on these real-world measurements and the periodic schedule. Results confirm the energy-efficiency of Janus; even with 9 neighbors *continuously* in proximity, the extreme case in our scenario whose maximum expected number of nodes is 10, our tag lasts 9.7 days in a reactive configuration and 48 days with a logging one. Further, these estimates assume 24-hour operation. In contexts where tags are worn only during working hours and switched off otherwise, lifetime obviously increases significantly, e.g., threefold for an 8-hour workday.

2.7 In-Field Experiments

2.7.1 Janus in Action: Experiences with COVID-19 Social Distancing

The previous sections offered a qualitative evaluation of Janus while in the next we depict the results achieved through experimentation in a practical context. The outcome of thesis experiments is twofold: first, they demonstrated the flexibility of Janus for several applications, from reactive to logging ones. Second, the data

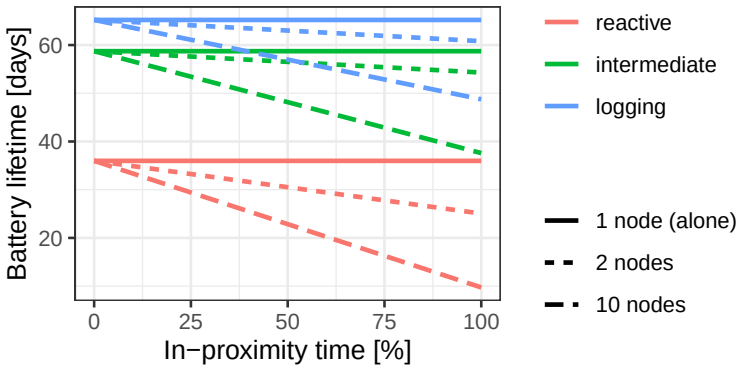


FIGURE 2.14: Estimated battery lifetime for a Janus tag vs. the percentage of time spent in proximity as a function of the time ratio spent in communication range with one or nine other devices.

collected and then analyzed provided the basis to use Janus in a less controlled complete study, as detailed in the next Chapter (3).

The COVID-19 pandemic offered several opportunities for experiments focused on social distancing, as workers in Italy were slowly returning to their duties with new safety rules in place. As mentioned (1) these social distancing scenarios combine both reactive and logging use cases, with the former specifically supported in our custom tag (§2.5) by audible alarms triggered when a safe distance is violated between two users.

In comparison to the previous experiments, here we did not have the possibility to gather detailed logs as in our tabletop experiments or have accurate ground truth as in our motion capture ones, due to the lack of infrastructure and memory limitations. Therefore, the goal is not to provide another in-depth system evaluation, rather to distill additional lessons learned from the in-field use of Janus, including practical aspects concerned with its configuration for different system scales and application requirements.

In all experiments described here, proper procedures were followed to recruit participants, compliant with GDPR and host organization regulations.

2.7.2 Before the Experiments: Configuring the BLE TX Power

The real-world experiences we report here are also the opportunity to comment about another configuration knob available that, although not specific to Janus, is of practical relevance in determining its performance: the BLE transmission power. A low-power setting reduces energy consumption; further, it may be preferable in dense scenarios with several people. In contrast, a longer range would discover many far-away neighbors whose irrelevant presence would trigger unnecessary distance exchanges, wasting both energy and memory resources. On the other hand, a low-power setting may yield insufficient reliability in scenarios where the wireless signal is likely hampered by obstacles, e.g., an industrial environment. Ultimately, the BLE TX power must be set by considering not only the tradeoffs between range and energy consumption but also the target application and its environment.

To this end, prior to the experiments reported here we analyzed the reliability of our BLE hardware with dedicated experiments spanning all TX power levels (-40 dBm to 4 dBm) across distances relevant for our application domain (1 to 10 m). We performed this analysis in an office corridor, representing a good balance among the target environments described next, and determined the ratio between the number of BLEnd advertisements collected during the experiment duration vs. those expected based on the configuration. We ascertained that all TX power levels > -16 dBm enabled correct reception of $>$90% of the advertisements at distances $<$2 m, relevant for our domain of social distancing. Nevertheless,

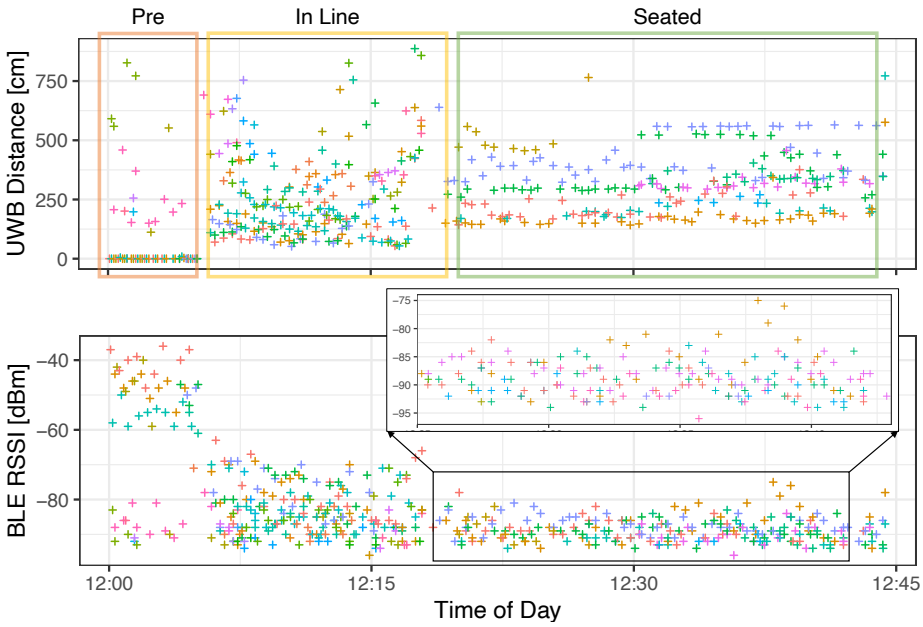


FIGURE 2.15: Cafeteria: Raw data from one individual. The zoomed-in area shows detail of BLE data.

hereafter we use different TX powers (-16, 0, and 4 dBm) precisely to cater for the different application and environment requirements, described next.

2.7.3 Cafeteria: Comparing BLE vs. UWB Raw Data

We begin with a campaign in a company cafeteria where, over a 2-hour period, we handed 90 workers a tag to carry during lunch. This dense setting is challenging both to discovery and ranging. However, the inherent flexibility of Janus allowed us to accommodate its scale, which is significantly higher than the one explored in the system evaluation (§2.6.1), by using the logging configuration with the same target parameters in Table 2.1 except for $N = 96$ and the consequently different and optimized BLEnd parameters. Moreover, considering that the wide open cafeteria area offers good radio signal, we used a BLE TX power of -16 dBm, the lowest among the acceptable ones we identified above, to reduce the number of discoveries far beyond the distance of interest, therefore improving energy and memory consumption.

Overall, 148,768 samples ⟨userID, RSSI, distance, timestamp⟩ were collected. Figure 2.15 shows the raw data of a single tag; each point denotes a measurement with a nearby tag, itself distinguished by color.

The data clearly shows three phases: when the tag is ready to be handed to the volunteer (*Pre*), when the latter is waiting to be served (*In Line*), and when the volunteer is eating (*Seated*). Nevertheless, while the distances between seated users are easily discerned in the raw UWB data (top), this is not the case in the BLE data (bottom), even when zoomed in to reveal detail; the latter vary significantly and continuously, while the former exhibit very clear and stable trends. Additional processing of RSSI values could improve matters, as done by many BLE-based approaches; however, this observation emphasizes that the raw, yet accurate data provided by UWB is already *immediately useful*.

2.7.4 Same-office Co-workers: Exploiting Raw Data

We report data gathered from a typical office area where the 7 members of a research group are physically co-located. We used the exact same intermediate configuration in Table 2.1. Further, we used a BLE TX power of 0 dBm, as this provides

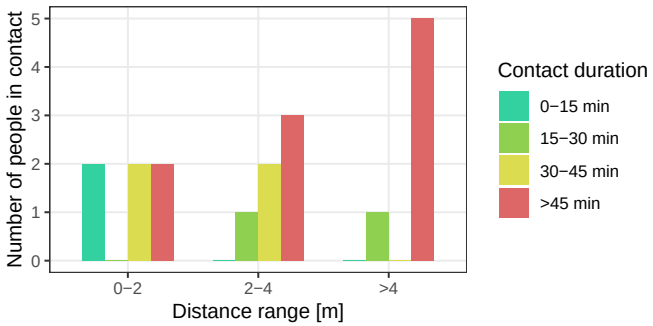


FIGURE 2.16: Small office: Cumulative time of one individual at given distance from others during one workday.

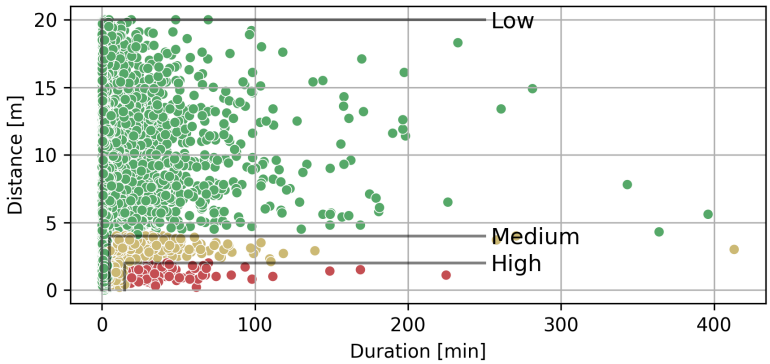


FIGURE 2.17: Company-wide: Contacts of 90 individuals over 3 days.

a good balance in this scenario with few people but several obstacles. Figure 2.16 shows the *cumulative* time one member spent near others during one day, and highlights a potential problematic situation: a significant amount of time (>45 minutes) was spent very near (<2 m) two other members, and only slightly less (30–45 minutes) very near two others. These times are derived straight from raw data, by simply summing the 15 s periods where a detection occurred. As such, they do *not* necessarily represent a (dangerous) *continuous* contact, whose definition we explore next. Nevertheless, this further emphasizes that the accurate raw data provided by Janus already offers actionable insights.

Interestingly, when we shared our analysis and raw traces with the volunteers they easily and promptly identified and recalled elements of their workday, e.g., meetings, lunches, and working as a pair on a project.

2.7.5 Company-wide: Using a Higher-level Contact Definition

We now show results from an aggregation of the raw Janus data into a higher-level notion of *continuous contact*, often used to characterize the risk of infection. We use the common definition of risky contact as one occurring for at least 15 minutes between individuals within 2 m. We process raw data sequentially, looking at all distance measurements between two individuals. We *open* a contact when we first find a value within threshold, plus a small tolerance (20 cm) accounting for measurement inaccuracies. We *close* the contact when this condition becomes continuously false for a given time period (90 s); the last value within threshold remains part of the contact. The overall duration t and average distance d of the contact is then computed, enabling a classification of contacts based on their risk: *i) high* when $d < 2$ m for $t > 15$ minutes, *ii) medium* when $2 \leq d \leq 4$ m for $t > 15$ minutes or $d < 4m$ for 5 minutes $< t < 15$ minutes, *iii) low*, otherwise. Although somewhat

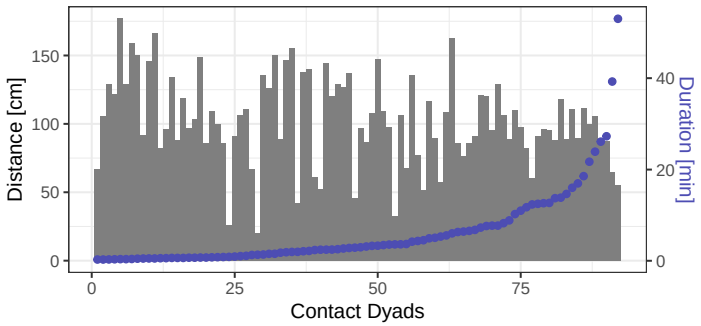


FIGURE 2.18: Factory floor: average distance and total time in contact over a 24-hour period for each reported contact dyad.

arbitrary, this classification is a realistic example of how contact data could help prioritize action.

To illustrate its power, enabled by Janus, we report 3 days of data collected with the intermediate configuration (Table 2.1) and $N = 90$ workers in a large company building. The office buildings were sparsely populated, therefore we chose the same BLE TX power of 0 dBm used in the previous experiment. Figure 2.17 shows the duration vs. distance of contacts, color-coded according to risk, providing a highly informative bird's-eye view. Overall, a total of 5,899 minutes were recorded in high-risk contacts over the 3 days. Although this seems large in absolute, on average it is only 21.8 minutes per person per day, about the same time users in the cafeteria scenario spent seated at lunch, potentially at high-risk distances. Longer accrued times were recorded at medium ($14,936$ minutes) and low ($77,659$ minutes) risk.

One can easily imagine follow-up analysis of this data, e.g., identifying the high-risk individuals, or analyzing the trends of risky contacts throughout the day. Fixed tags throughout the building (e.g., at coffee machines) could also provide approximate locations for some contacts.

2.7.6 Factory Floor: Real-time Alerting and Contact Tracing

We conclude by presenting data from 30 tags used by a company on a factory floor. In this case, we used the highest BLE TX power of 4 dBm, to cater for relatively few workers over a rather large area with several obstacles to communication, due to the industrial environment.

The focus here was real-time alerting; tags used the corresponding reactive configuration (Table 2.1), although some amount of logging was also supported for later analysis. Specifically, tags were programmed, as part of full-fledged product integrating Janus via its API (§2.5), to record only high-risk contacts and offload them opportunistically on-the-fly to nearby gateways connected to the cloud, another functionality enabled by accurate and energy-efficient proximity detection.

We focus our attention on pairs (*dyads*) of individuals, and their total contact time in a day (Figure 2.18). If tags A and B were within 2 m for 6 minutes in the morning and 9 minutes in the afternoon, the chart shows a point for dyad A–B at 15 minutes, with the corresponding histogram showing the average distance of the dyad. For 30 individuals, there are 435 possible dyads; however, only 92 (21%) were reported in contact. Of these, only 9 dyads exceed 15 minutes of total contact time. Further, these involve only 13 distinct tags, suggesting that long contacts are concentrated in few individuals; this is expected based on their duties in the factory, e.g., cooperatively moving heavy objects.

2.8 Discussion

The combination of system evaluation (§2.6.1) and real-world experiences (§2.7.1) allowed us to evaluate Janus with increasing degrees of complexity, covering all the requirements and design goals (§2.1) while simultaneously grounding them in the needs of real-world applications. For instance, although the scale of our system evaluation is limited to 10 nodes for logistical reasons, our experiences bear witness to both the scalability and flexibility of Janus: our first reported experience in the cafeteria was configured for a setting with an order of magnitude more nodes than used in our system evaluation.

On the other hand, our analytical model (§2.3), validated through system evaluation, also shows an inherent tension between scale and latency. Our reactive configuration, with a discovery latency of $\Lambda = 2$ s, was used on over 2,000 tags across several factories for social distancing and proximity warning applications, showing that Janus can be used successfully in large, dynamic environments. Nevertheless, a combined increase in scale and decrease in latency may prove infeasible beyond a given point, because *i)* accommodating many ranging targets requires long epochs, while a low latency requires short ones, and *ii)* the increased density of competing ranging windows increases the likelihood of collisions.

For similar reasons, greatly reducing the value of Λ, even at a small scale, becomes problematic beyond a given point, e.g., if sub-second latency is required to detect proximity with very fast moving targets. While decoupling discovery and ranging latency (§2.2.1) may help, scenarios with very low latency were not considered as part of those motivating Janus (§2.1) given that frequent ranging is inherently at odds with energy efficiency. These types of scenarios are explored in the work of Chapter 4, with the Hermes protocol.

Ultimately, not only is there no one-size-fits-all configuration, but also the various performance dimensions are intertwined, making it difficult to determine how much Janus scales or how fast detection can be in purely abstract terms. Instead, we put forth a contribution of practical relevance by relying on the predictable operation of Janus to provide domain experts with a small number of knobs to identify a good configuration meeting their application and system requirements. We concretely exemplified this flexibility and configurability in several paradigmatic applications.

In this respect, we have also shown that the impact of a misconfiguration of the maximum expected number of nodes (e.g., twice the size in our system evaluation) is far from dramatic. From a system standpoint, this confirms one dimension of scalability. From a practical standpoint, this is important because, unlike the detection latency Λ or the probability of discovery P_d which are often clearly defined by the application, the domain expert may have only an educated guess about the worst-case system scale N, especially when targeting contexts with humans or wildlife whose social interaction characteristics are precisely the unknown to be investigated.

While Janus offers additional opportunities for customization to application needs, they remain beyond what we could address here. For instance, the decoupling between neighbor discovery and ranging latency offers different tradeoffs from those required by our target applications, notably improving reactivity when needed.

Moreover, other opportunities come directly from the underlying BLE layer. We have touched upon the need to properly configure BLE range when discussing our real-world experiences (§2.7.1). Interestingly, the BLE range can also be exploited to improve detection of fast-moving objects; a long range enables their detection when still far away, guaranteeing subsequent timely distance estimation. The improved communication range of the recent Bluetooth 5 (up to 4x w.r.t. BLE [23]) may be an asset in this respect. Moreover, the BLE *scan-response*, in which the scanning node can reply immediately to the advertising one, could also in principle be exploited to significantly decrease the latency of bidirectional discovery. Nevertheless, this would increase both collisions and energy consumption in

BLEnd, due to the increase in transmissions and receptions, and ultimately invalidate the predictable (and validated) performance guarantees we rely on. Instead, the Janus API already provides developers options to exploit PHY-level information for application-specific needs. For instance, the RSSI of advertisements can be exploited as a coarse estimate of distance. This can either be used by the application before the accurate UWB range is determined, or used inside Janus to limit ranging only to "near-by" neighbors, providing an additional knob to navigate the tradeoffs between communication range and scalability.

Dual Radio Architectures - A Comparison

This dual-radio approach is largely novel in both research and commercial devices. Among the latter, a few [81, 97] use BLE only as an out-of-band channel to collect data and set configurations. Research works instead exploit the two radios in the very mechanics of proximity detection.

The work in [55] describes a system for monitoring the elderly exploiting *both* the RSSI information in BLE advertisements *and* the accurate distance returned by UWB. This enables a design where UWB ranging is exploited at lower rate than normal, saving energy, and accuracy is improved w.r.t. BLE alone, albeit lower than pure UWB. Therefore, although the two are used in synergy, the goals and outcome are different w.r.t. Janus, which achieves high energy efficiency *without* sacrificing accuracy. Further, the proposed system is infrastructure-based, as it relies on constant communication against anchors with both radios. Nevertheless, a similar technique in principle could be a valid complement to the current design of Janus, unlocking new tradeoffs between energy, accuracy, and other requirements (§2.1). For instance, the advertisements sent by BLEnd could help estimating distance when a UWB-based one is not available, as in between the first discovery and first ranging, or upon collisions.

Instead, the very recent SociTrack system [10] exploits BLE for neighbor discovery, via the BLEnd protocol also incorporated in Janus. This is however exploited *only* as a sort of "trigger" for the UWB layer, which is therefore responsible not only for ranging exchanges, but also for their coordination; this is achieved with a centralized, network-wide flooding, which further increases the energy burden. Like Janus, SociTrack decouples in time the operation of BLE and UWB, but activation of the latter is further delayed by global schedule dissemination. Unfortunately, its effect on latency of first ranging is not evaluated in [10], unlike the analysis we provided (§2.6.1).

On the other hand, a higher accuracy is reported for SociTrack, thanks to a specialized triple-antenna tag design exploiting spatial diversity. However, its accuracy is not evaluated in-depth in mobile scenarios, let apart with accurate ground truth (§2.6.4). Instead, we are limited by the popular off-the-shelf, single-antenna, dual-radio DWM1001C module, which nevertheless makes our firmware design immediately applicable to the many research and commercial systems based on it.

Moreover, the multiple packets necessary in SociTrack to exploit spatial diversity, combined with their global UWB-based coordination, yield significantly higher energy consumption w.r.t. Janus. Indeed, the authors highlight a 12-day lifetime on a 2000 mAh battery. This is obtained with an update interval $U = 300$ s and a "network" of only $N = 2$ nodes; with $U = 2$ s and $N = 10$, lifetime decreases to ∼4 days. In these latter conditions, Janus achieves 10 days (Figure 2.14) and with a battery of only 950 mAh—i.e., a 2.5x lifetime with a battery half the size. With a longer update rate of $U = 30$ s Janus achieves 50 days, i.e., 4x more than the best result of SociTrack, with an update rate 10 times higher and 5 times more nodes in continuous range.

2.9 Concluding Remarks

In this chapter we described Janus, a novel dual-radio network protocol enabling

accurate and energy-efficient proximity detection. Janus embodies a novel design exploiting BLE for discovery devices in range and coordinating their ranging exchanges, performed via the UWB radio without the need for infrastructure. We proved that this scheme can be implemented efficiently, modeled its reliability, evaluated experimentally its performance in reference configurations for paradigmatic use cases, and reported about in-field experiences concretely showing its practical relevance. More importantly, we confirmed that Janus is accurate *and* energy-efficient, achieving a mean error of at most $\sim$30 cm while enabling weeks to months of uninterrupted operation depending on the use case.

On the other hand, our realistic experiments with people wearing tags in a motion capture facility offering mm-level ground truth also evidenced that the relative antenna orientation between devices and the signal occlusions caused by the human body significantly affect ranging accuracy, generating errors well above the decimeter-level ones typically associated with UWB. As the latter has been used mostly for localization these issues have been largely neglected, but are crucial for proximity detection, where devices are typically worn by people.

Chapter 3

Measuring Close Proximity Interactions with Janus

Policy makers have implemented multiple non-pharmaceutical strategies to mitigate the worldwide COVID-19 crisis. Interventions had the aim of reducing close proximity interactions, which drive the spread of the disease. Deeper knowledge of such physical, human interactions has become necessary, especially in settings involving children whose education and extra-curricular activities should be maintained as much as possible at pre-pandemic levels. Despite their relevance, almost no data are available on close proximity contacts among children in schools or other educational settings during the pandemic.

After an introduction on close proximity interactions with a review of related works, this chapter describes how the Janus protocol has been used to collect real contact data of children and educators in three summer camps during summer 2020 in the province of Trento, Italy. In these scenarios, the wide variety of daily activities induced multiple individual behaviors, allowing a rich investigation of social environments from the contagion risk perspective. We considered risk based on duration and proximity of contacts and classified interactions according to different risk levels.

We then used the collected data to evaluate the summer camps' organization, observe the effect of choice to partition the children into small groups, or social bubbles, and identify the organized activities that mitigate the riskier behaviors.

The results presented here demonstrate the potential of proximity detection protocol, highlighting the importance of research on this field.

The work presented in this chapter has been published in [60].

3.1 Introduction

Close proximity interactions (CPIs) drive the spread of any disease that is transmitted predominantly by respiratory droplets and saliva, such as influenza, common colds, and severe acute respiratory syndromes (i.e., Severe Acute Respiratory Syndrome (SARS), Middle East Respiratory Syndrome (MERS), Coronavirus Disease 2019 (COVID-19)) [12, 87, 35, 92, 46, 61]. An improved characterization of CPIs should thus lead to a better understanding of the spread dynamics and possibly inform public health experts and policy makers to design more effective interventions [102].

For this reason, some research efforts have used wearable devices and RFID or infrared (IR) sensors to measure and analyze high-resolution proximity interactions in different settings such as schools [92, 94], workplaces [18, 4], hospitals [50, 98, 43, 29, 28], households [77], and conferences [**cattuto20:plosoneisella2011j**, 95].

During the COVID-19 pandemic, social contacts and in particular CPIs were significantly modified [51, 108, 33, 109] by several non-pharmaceutical interventions such as physical distancing measures (i.e., 1 m or more), mobility restrictions, closings of schools, universities, and selected businesses (e.g., restaurants, bars, coffee shops, gyms), promotion of teleworking, cancellations or limits on the size of events (e.g., sports events, weddings, funerals), limits on the number of people

in small family, educational and social gatherings (i.e., social bubbles), etc. [42, 45, 13].

However, despite their relevance, almost no data are available on how CPIs occur among children in contexts such as schools or summer camps during the COVID-19 pandemic, thus making it difficult to evaluate and model the effects of physical distancing measures, small group strategies, preferences for outdoor activities, masks, etc., on CPIs, as well as identifying the situations and activities during school and summer camp days where the risk of transmission is elevated.

The collection of reliable data in these environments (e.g., schools, summer camps) is itself a nontrivial task. During the pandemic, several local and national governments have launched smartphone digital contact tracing (DCT) apps based on the BLE technology [37] and the GAEN (Google and Apple Exposure Notification) interface [39], and several studies have shown the effectiveness of Bluetooth-based DCT using real-world contact patterns [19, 72] and in pilot and country-wide studies conducted in Switzerland, the United Kingdom (the Isle of Wight and the whole country), and Spain (Gomera island) [91, 54, 89, 105].

In addition to the challenge that most children do not carry personal smartphones, this technology has at least two shortcomings for capturing CPIs in schools and summer camps: (i) *low temporal resolution* (e.g., GAEN detects neighbors every 4 minutes [39]), and (ii) *low spatial resolution*, which directly descends from limitations of BLE and leads to significant estimation errors [58]. The first issue can be tackled by the use of an alternative to GAEN, while the second can be addressed by changing the technology used for estimating distances, e.g., to ultra-wideband (UWB), which brings the spatial error down from meters to decimeters [107].

In this chapter, we show how we address these issues using the Janus protocol described in the previous chapter. In our experiments, we configured Janus to acquire distance measurements every 30 s and installed it on a wearable device that children can easily carry. We have collected real-world CPIs with Janus at three summer camps in the province of Trento (Italy). These camps offer interesting settings because of the rich variety of daily activities that induce different CPIs among children and between children and the summer camps' educators. Moreover, the summer camps took place during the summer of 2020, in the middle of the pandemic and just after the local easing of lockdown measures. As such, it is possible to investigate the effect of the guidelines and regulations enforcing physical distancing, mask-wearing, outdoor activities, and the formation of small groups (i.e., social bubbles).

The accurate and fine-grained contact data uniquely enabled by Janus, complemented by the metadata about summer camps, results in the rich data set that is the basis of the presented multi-level analysis. First, we describe the definition of *close contact* as the aggregation of multiple raw measurements captured by the sensors and discuss the modeling choices implied by this operation. After this aggregation phase, the resulting contacts are enriched with metadata. For example, social bubbles [11, 59] were enforced as a contagion containment measure, and thus we assigned to each contact the groups of the two involved individuals. Further, each contact is associated with the activity being performed during the contact time.

By considering the metadata in the analysis along with the raw contact data, we offer novel insights into both educator-child and child-child social interactions during the pandemic. In particular, we study the distribution of the level of contagion risk among individuals depending on the proximity and duration of their contacts, finding that a vast majority of CPIs are classified as low risk. Moreover, we aggregated the contacts as intra-group (i.e., within the social bubble) and inter-group (i.e., between different bubbles), and observed changes in the distribution of contact risk levels in the two cases, offering evidence of the effectiveness of the social bubble strategy [11, 59]. Finally, a thorough analysis of the different activities provides insights into their inherent risks of contagion, which can be further interpreted in view of the features of the activity itself (indoor or outdoor, static or dynamic, etc.).

TABLE 3.1: Description of the three summer camps investigated in our study.

ID	Short Description	Ages	Children	Educators	Groups
AM-PRI	Morning camp with a large indoor space, nearby a public park.	6-11	21	5	3
DAY-PRI	All day camp in an alpine region with only outdoor space.	6-11	13	5	2
DAY-INT	All day camp in an alpine region with additional indoor space.	11-14	9	2	1

3.1.1 Modeling the spread of infectious diseases from proximity data

The usage of close proximity data in the state of the art is crucial to understand the novelty of the work presented in this chapter. Therefore this section presents various works on modelling the spread of infectious diseases with proximity data.

The analysis of proximity contact data includes multiple works that focus on modeling the spread of infectious diseases. For instance, [92] have used sensors with a proximity resolution up to 3 meters in a high school to obtain a dataset in which they have simulated the spread of an influenza-like disease. Doing this, they have found results in agreement with absentee data during the influenza season.

Another line of studies has exploited data collected in different environments within the SocioPatterns project. In particular, the estimation of face-to-face interactions was used to correct theoretical epidemiological infectiousness parameters and thus to obtain a better risk estimation for generic spreading processes [34, 9, 36, 66, 98], or to identify specific individual roles in workplaces, hospitals, schools that would be more responsible for the spread of a disease [77].

Other face-to-face interaction data have been collected by [29] to understand how hospital-acquired infections spread and to possibly design control strategies. Similarly, [75] collected CPIs and data about a staphylococcus transmission in a hospital, finding that collected CPIs were able to correctly reproduce transmissions and thus demonstrating the importance of this tool to trace disease spread.

Additional studies have also focused on contact tracing strategies, such as the work by [32] and more recently, for COVID-19, the ones of [19] and [72].

The study presented here describes the collection of real-world daily educator-child and child-child close proximity interactions at three summer camps during the COVID-19 pandemic. This unique dataset allowed us to characterize contagion risks based on duration and proximity of contacts and classify interactions according to different risk levels. We then investigated the effect of the guidelines and regulations enforcing physical distancing, observe the effect of partition in small groups (i.e., social bubbles), and identify the summer camp activities that mitigate the riskier behaviors in terms of contagion.

3.2 Materials and Methods

Here, we offer details about the summer camps where the in-field experiments were performed and the mechanics of data acquisition, and state the definition of close proximity contact used throughout the chapter.

3.2.1 Data acquisition

The data used in our analyses results from a study conducted from August to September 2020 in three different summer camps, summarized in Table 3.1, in

TABLE 3.2: Daily activities at the summer camps, each with a brief description, the location and the duration in minutes for each summer camp that offered the activity.

Activity	Description	Location	AM-PRI	DAY-PRI	DAY-INT
Woods	Playing in a wooded area	outdoor	90 min		
Soccer	Playing in a soccer field	outdoor	90 min		
Board games	Playing tabletop games	indoor	90 min		
Newspaper	Pairs work at computers	indoor	90 min		
Theater	Singing and acting	indoor	90 min		
Snack	Short food break	indoor	15 min		
Team games	Organized group games	indoor	90 min	120 min	120 min
Crafts	Arts and craft	indoor	90 min		180 min
Hiking	Group walk	outdoor			240 min
Round table	Greetings, planning, etc	indoor			180 min
Day closing	Free play pre pick-up	outdoor		30 min	60 min
Outdoor lunch	Eating	outdoor			60 min
Indoor lunch	Eating	indoor		60 min	
Free play	No organized activities	indoor			60 min
Free play	No organized activities	outdoor		60 min	

Trentino, Italy. The study design was approved by the Agency for Family, Birth, and Youth Policies (Agenzia Provinciale per la Famiglia, la Natalità, e le Politiche Giovanili) of the Autonomous Province of Trento[1], the provincial government body responsible for the organization of the summer camp programs, and by the two social cooperatives directly responsible for camp management and activities.

In preparation for the study, parents and educators were provided with detailed information about the purpose of the study, the data treatment and privacy enforcement strategies, the devices the children and educators would be using, and the measurements they provide. Following Italian regulations, all parents and educators signed an informed consent form. Special attention was given to privacy and data protection: no personal information was associated with the identifier of the corresponding Janus device. We did note the group (i.e., social bubble) the individual belonged to and, in some cases, the identity of devices carried by others for whom physical distancing rules were waived (e.g., among siblings and between children with special needs and the educators assigned to assist them).

The first summer camp, AM-PRI, operated for half days (mornings) with 21 primary school-age children and 5 adult educators, all of whom agreed to participate in the study. The children were divided into 3 groups, each with one or two educators. Each activity during the day was restricted to a single group at a time to maintain separation and leverage the concept of social bubbles [11, 59].

The second and third camps were organized the same week by the same cooperative, but took place at different locations; therefore, we treat them separately. Both were all-day camps from 8:00 to 16:30. DAY-PRI applied the social bubble with two groups of primary school children. The third camp, DAY-INT, involved 9

[1] https://www.trentinofamiglia.it/

FIGURE 3.1: Janus device management at the AM-PRI camp. Left: An educator fitting the waist bag containing the device on a child, on the first camp day. Right: Devices in waist bags sitting on a storage bench overnight; the inhibitor device is inside the red bag in the center.

intermediate school children with two educators. The overall participation rate in these two camps was 94%.

The summer camps engaged the children in different educational and playing activities, as summarized in Table 3.2. For each activity, we indicate the approximate duration in minutes for each camp.

Device setup and experimental setting

To make carrying the device comfortable for the children, we inserted it inside a waterproof waist bag, as shown on the left of Figure 3.1. We received positive feedback from the educators, who said that the children immediately forgot they were wearing the device. The Janus device was configured to sample distances every 30 s when devices are in proximity. Measurements greater than 10 m are discarded to save memory on the device and because these large distances are not considered relevant for the transmission of SARS-CoV-2 [52, 22].

After programming the devices and inserting new batteries, the waist bags were delivered to camp organizers at the beginning of each week. The educators were responsible for handing out the bags to the same children each morning and collecting them at the end of the day. At the end of the week, the devices were collected and the data offloaded via Universal Serial Bus (USB).

As the devices do not have an on/off switch, to avoid the collection of meaningless data at night, when devices were stored on a bench (Figure 3.1), we implemented an *inhibitor* device. This special device was turned on at the end of the day by connecting it to a USB power bank. When the regular devices detected the BLE advertisement of the inhibitor, they went to sleep for 5 min. Upon restarting, if the inhibitor was detected again, they returned to sleep; otherwise, they started functioning normally, ranging with all neighboring devices. Each morning, the inhibitor device was detached from its power supply. This inhibition mechanism saved battery as well as memory and, most important, required no technical skills from the educators; even using the USB power bank was much easier than removing the battery from all Janus devices, which was the only other alternative available.

Pre-processing of the data

Prior to analysis, the data collected during each summer camp were cleaned of spurious samples recorded by the devices. We describe the process here and report a summary of the collected data for each setting.

The Janus devices do not have an on/off switch as mentioned in the previous section, and as a result, are active 24 hours per day, not only when the summer camps are in session. Although we used the inhibitor device to limit the measurements taken after the daily close of the summer camp, some additional measurements are still stored.

TABLE 3.3: Statistics of the raw data sets, including the number of measures before and after the pre-processing step.

ID	Initial day	Final day	Unique users	Raw measures	Filtered measures
AM-PRI	2020-08-17	2020-08-21	24	222222	48739
DAY-PRI + DAY-INT	2020-08-24	2020-08-30	25	213219	146576

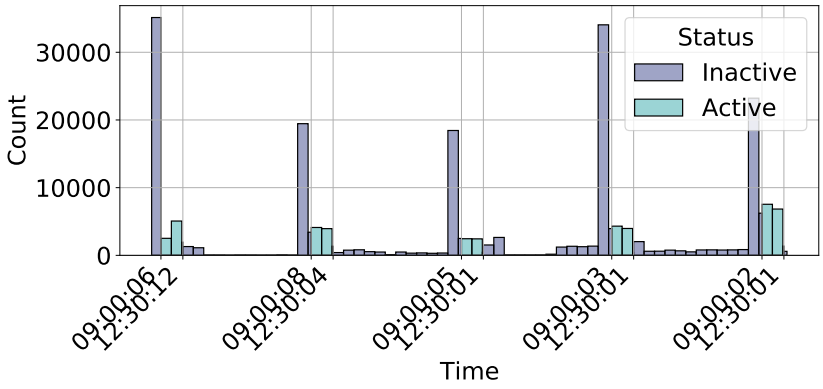


FIGURE 3.2: Filtering of the spurious measures. Distribution of the measurements over the entire sampling period, either with *Active* or *Inactive* status for AM-PRI.

For example, if the BLE signal to the inhibitor was weak, the devices may have been briefly activated. Additionally, the inhibitor node was often disabled several minutes before children arrival and devices distribution, resulting in measurements among the devices still on the storage bench. Finally, some children were absent for entire days or arrived late while their device was still taking measurements.

Identifying all these cases was a largely manual effort based on information from the educators about absences and observations in the data itself. For example, when a sequence of constant distance measurements is seen at the beginning of the day, it is likely that the devices are still in storage, as children are rarely so still. The data cleaning step filters all these spurious measures. Table 3.3 shows for each summer camp the data collection time frame, the number of unique participants that have been involved, the number of overall measures, and the number of measures after the filtering step.

Figure 3.2 shows the distribution of the entire measurement set for AM-PRI. The time intervals during which the activities took place (*Active*) are separated from the time between the activities (*Inactive*). The peaks of data close to the morning camp start time correspond to the phase when the inhibitor node is off, but the devices have not yet been distributed to the children. In this case, all devices are immobile, near one another on a bench (Figure 3.1) and thus save many distance measurements.

3.2.2 Definition of close proximity contacts

After downloading the measurements from all devices and pre-processing them, we aggregated these raw samples into *contacts* characterized by two device IDs, the timestamp marking the beginning of the contact, the contact duration, and a distance, as described next.

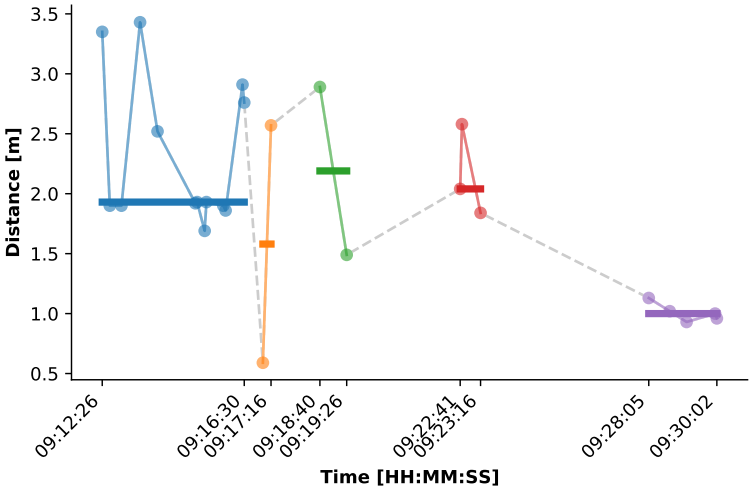


FIGURE 3.3: Measurement splitting and contact aggregation process. The figure shows the measurements collected in the first 20 min of August 8th, 2020, between node 26 and 27 at the AMPRI camp. The measurements (light colors) are colored according to the division into contact characterized by $\tau_{time} = 90\ s$ and $\tau_{space} = 2\ m$. Each contact is depicted as a horizontal bar from its beginning to its end, where the height of the bar represents the median distance.

To identify a contact, we focus on a pair of IDs, collecting all measurements captured by either device, and sorting them in time. This sequence is then processed sequentially to divide the time into multiple, meaningful contacts. Intuitively, a contact should contain measurements that are all *temporally and spatially close* to one another, which we define via time and distance thresholds.

We begin with the temporal dimension, splitting the sequence into sub-sequences whenever a gap of $\tau_{time} = 90\ s$ exists between two consecutive measurements. This step accounts for interruptions in the interaction between the pair of devices, e.g., when they move away from one another.

Second, we check each of the distances inside each sub-sequence, ensuring that a single contact contains only measurements with similar distances, and ensuring that the single distance attribute assigned to a contact has a reasonable spatial variation. Therefore, we sequentially process the measurements of a sub-sequence in temporal order, and retain them in a single sub-sequence as long as all the measured distances are within $\tau_{space} = 2\ m$ from each other; a new sub-sequence is started upon the first measurement outside this range.

In this way, we obtain a set of sub-sequences, each containing measurements without large temporal gaps and with similar distances. After discarding sub-sequences with fewer than $\tau_{len} = 2$ measurements, we aggregate each cluster into a contact. Each contact is tagged with the timestamp of the first measurement in the sub-sequence, a duration given by the time span of the measurements in it, and a distance given by the median value of the measurements. Using the median (i.e., the central value of the distribution) yields a more robust value compared to the mean, which is more sensitive to extreme values and outliers.

An example of this splitting and aggregation process is shown in Figure 3.3, which depicts a sequence of measurements in a 20 min period grouped into sub-sequences (identified by colors) and aggregated into contacts (identified by the horizontal lines). The different splitting strategies can be observed. For example, the orange and green sequences are separated due to the gap of more than τ_{time} between them. On the other hand, the blue and orange sequences are separated because the first measurement in the orange cluster is outside the range of τ_{space} with respect to the previous measurements.

TABLE 3.4: **Description of the contacts resulting from the aggregation procedure.** For each camp, we report the total number of contacts, the average number of the measurements for each contact, the number of groups and activities in the camp, and the number and percentage of the contacts that are uniquely associated with an activity. For DAY-PRI and DAY-INT, we report both the number of activities, and the number of activities considered for the analysis (in parenthesis).

ID	Num. Contacts	Average Measurements per Contact	Num. Groups	Num. Activities	Activity-tagged contacts
AM-PRI	7259	5.80	3	8	6833 (94.13 %)
DAY-PRI	7561	8.48	2	5 (4)	6774 (89.59 %)
DAY-INT	3485	16.40	1	9 (7)	3485 (100.00 %)

The resulting contacts model the high-level notion of CPI that we use in our analyses in the next sections, and enables the general contagion risk assessment of the different environments. Further, we also associate to each contact the groups of the involved IDs and an activity when both IDs are in the same group.

Some of the contacts can be removed a posteriori to account for risk-modelling choices. For instance, we discard contacts between siblings (who were not required to respect physical distancing rules) or between children with special needs and their support teacher. Additionally, in DAY-PRI and DAY-INT, the two activities "welcoming activity" and "swimming pool" have been discarded because the devices had not all been distributed and were piled up in the same place, resulting in many spurious measurements.

The resulting numbers of contacts for each summer camp setting are reported in Table 3.4. For each data set, we also report the number and percentage of contacts where both users belong to the same group, and thus to which we are able to assign an activity.

3.3 Results

Leveraging the previous definition of contacts and additional metadata, we can now delve into the analysis of the complex daily CPI patterns within the summer camps.

3.3.1 Identification of contagion risk levels

To build a general model for risk analysis, we define four different categories of contagion risk for contacts based on proximity and duration. We then classify all contacts into these categories.

In a meta-analysis and systematic review of observational studies on Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV), Middle East Respiratory Syndrome-related Coronavirus (MERS-CoV), and SARS-CoV-2 (Severe Acute Respiratory Syndrome Coronavirus 2) person-to-person transmission [22], a physical distancing of less than 1 m was reported to result in a significantly higher transmission risk than distances higher than 1 m (12.8% vs. 2.6%), thus supporting a minimum physical distance of 1 m, as in the rule enforced in schools and summer camps in Italy. However, as pointed out by [52], physical distancing rules would be more appropriate and effective if they offer graded levels of risk. Similarly, although contact tracing guidelines in several countries, various digital tracing contact apps, and some studies [20] assume that the duration of exposure to a person with COVID-19 influences the transmission risk (e.g., defining a threshold

TABLE 3.5: Risk levels of contagion defined on the basis of duration of exposure and physical distance.

	Duration	Distance
● High risk	$\geq$ 15 min	$\leq$ 1 m
● Medium high risk	$\geq$ 10 min	$\leq$ 2 m
● Medium low risk	$\geq$ 5 min	$\leq$ 4 m
● Low risk	$<$ 5 min	$>$ 4 m

of 15 min beyond which transmission risk increases), a precise quantification of the duration of exposure is still missing [52].

Following these considerations, we define the risk categorization summarized in Table 3.5. The first category is associated with a *high risk* of contagion and includes all contacts with duration above 15 min and distance less than 1 m. The second category, *medium-high risk*, includes all contacts with duration above 10 min and distance below 2 m that are not included in the high-risk category. The third category, *medium-low risk*, includes contacts with duration above 5 min and distance below 4 m not included in the previous categories. The fourth category contains all remaining contacts, therefore associated to a *low risk* level.

Notably, this granularity in discriminating risk levels is enabled by the fine-grained spatio-temporal resolution offered by Janus. The high accuracy of UWB ranging, in contrast to the coarse, RSSI based distance estimation [110] with errors on the order of meters, enables spatial discrimination at the granularity of a meter. Similarly, our configuration of Janus captures distances every 30 s, while the popular GAEN interface collects a single sample in each 4 minute window.

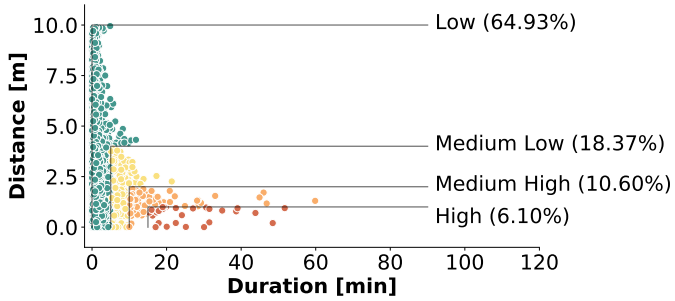
It is worth noting that while our data is rich in terms of accuracy, the Janus platform does not capture whether interactions are face-to-face. However, while face-to-face interactions provide a good approximation of conversations and are useful for social interaction analysis [74, 18], when studying SARS-CoV-2 transmission this aspect is less critical. Indeed, several researchers are highlighting that SARS-CoV-2 can spread among people occupying the same space, whether or not they are facing one another [71, 41].

Further, while our definition of risk level is context-agnostic, based only on proximity and duration in line with the national and international policy recommendations [31, 30], our analysis in the following sections is context-aware as it takes into account metadata that notably includes whether or not the contacts occurred indoor or outdoor. This two-step strategy allows for an in-depth risk assessment and effective definition of the risk levels without requiring possibly intrusive and privacy-critical contextual information.

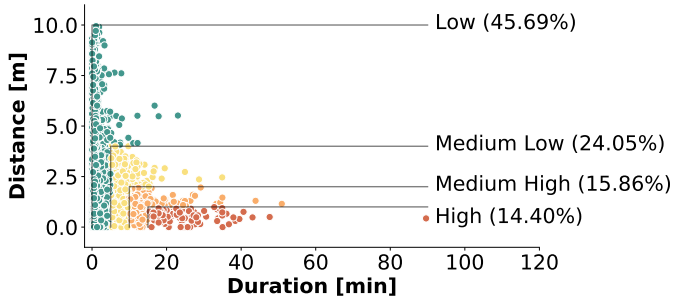
3.3.2 Contagion risk analysis

Figure 3.4 shows a scatter plot for each summer camp dataset, reporting the recorded contacts as a function of duration and proximity. Each dot represents a contact, as defined in Section §3.2.2, with colors describing the associated risk according to the color code in Table 3.5. The percentages reported inside the figures, and associated with the different risk levels, represent the percentage of time spent by the population in the corresponding risk category. Interestingly, we see that, even if different summer camps imply different levels of risk, there is a non-negligible percentage of contacts at high risk of contagion in all summer camps.

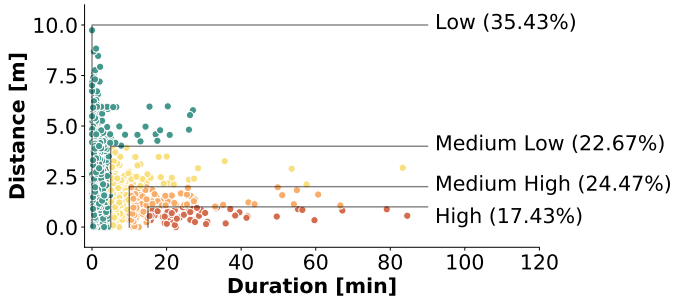
In the representation in Figure 3.4, each dot represents a single contact between two individuals, but it ignores information about the corresponding IDs. Therefore, it is possible that the analyzed population has heterogeneous behaviors, e.g., with only a few participants involved in more risky close proximity interactions and the majority of individuals interacting safely, or vice-versa. To understand how the risk is distributed among the summer camp population we consider three additional views, shown in Figure 3.5, where we examine the behavior for pairs of individuals. We report only the case of AM-PRI, since the other camps yielded



(A) AM-PRI

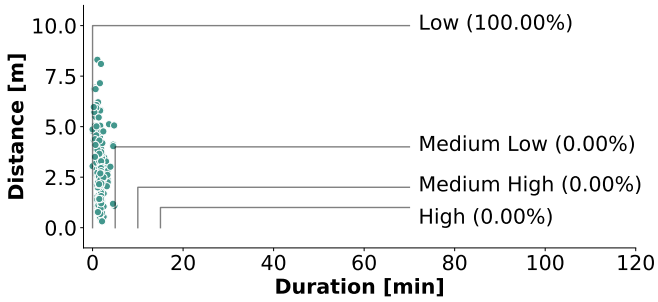


(B) DAY-PRI

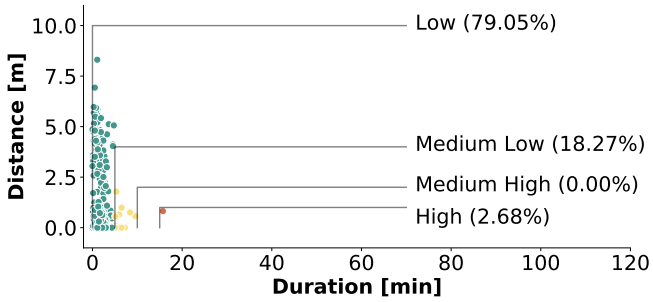


(C) DAY-INT

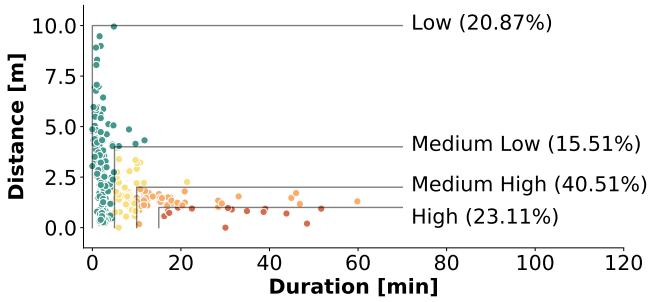
FIGURE 3.4: Summer camp contacts and contagion risk. The figure reports, for each summer camp, the corresponding contacts classified according to their risk of contagion as a function of the duration of exposure and proximity, following the risk categories in Table 3.5. The values in parentheses denote the percentage of time spent in a contact with the corresponding risk category.



(A) Mean duration of exposure and mean spatial distance.



(B) Minimal spatial distance.



(C) Maximal duration of exposure.

FIGURE 3.5: Unique contacts and risk levels. Contacts from AMPRI, aggregated into a single point (one per device pair) according to different criteria.

analogous results. First, in Figure 3.5a, we compute for each pair the average distance and duration across all the contacts, resulting in a single dot per pair. We observe that each pair interacts, on average, in low-risk social interactions. A similar result is observed in Figure 3.5b, where we select the single contact per pair with the smallest proximity distance. Finally, Figure 3.5c shows the single contact per pair with the longest duration. Here, we see that ∼23% of the pairs of individuals are involved in very high-risk interactions. From this, we conclude that the risk of contagion is distributed quite homogeneously among the different pairs of individuals, except for some for which the longest interactions are also the most dangerous ones.

These graphical representations give a first, general idea of the contact risk levels and offer an understanding of how the risk is distributed among the individuals. We note that these analyses depend on our definition of contact and, particularly, on the thresholds defined in Section §3.2.2.

In addition, the proposed contact definition allows us to perform two types of meta-analysis based on the risk levels related to: (i) group dynamics (e.g., CPIs

TABLE 3.6: **Summary of the number and duration of the contacts in the three camps according to the social bubble strategy.** For each camp AM-PRI, DAY-PRI, and DAY-INT, we report for the different bubbles the total time of contact and the number of contacts organized by the role of the participants.

		Intra-group			Inter-group		
		child child	child educator	educator educator	child child	child educator	educator educator
AM-PRI	Time [min]	10362.40	2285.28	77.82	462.12	181.77	11.75
	Number	5484	1297	52	295	121	10
DAY-PRI	Time [min]	12075.02	5250.60	290.32	538.95	383.93	72.33
	Number	4388	2064	145	341	195	49
DAY-INT	Time [min]	7732.58	2229.58	4.22	-	-	-
	Number	2004	613	2	-	-	-

among group members, among members of different groups, educator-child interactions, child-child interactions), and (ii) the type of educational and recreational activities planned during the summer camp.

As described in Section §3.2.1, each summer camp setting organized participants in small groups and in specific roles (educator, child). Groups are intended to keep participants separated into disjoint bubbles [11, 59] so that any contagion event would remain localized. On the other hand, roles reflect the internal organization of the summer camps, where both users (children) and educators (adults) were present. The results are graphically reported in Figure 3.6, where the colored bars show the relative percentages of contacts for each risk level that can be attributed to child-child, educator-child, and educator-educator interactions, respectively. Moreover, these can be divided into interactions involving two people belonging to the same group ("intra-group") and those bridging two different groups ("inter-group"). Instead, the large grey bars in the background report the total percentages of contacts for each specific type of interaction, independently on the associated risk. To facilitate the quantitative comparison of the results, Table 3.6 reports, for each summer camp, the number and the total duration of the contacts in the six groups.

When a contact occurs between two members of the same group, we assign to it the activity being performed at that moment by that group. In this way, we add another layer of analysis that allows us to study the relationship between the activity type, the number, and the contagion risk level of the contacts. The results are shown in Figure 3.7, where we report four bars for each activity, representing the four risk levels. The height of the bars represents the sum of the duration of all contacts during each activity divided by the total duration of the activity. Hence, each bar reports the risk per unit time of each activity. This normalization allows comparison across the different activities, independent of their duration. The percentages show the fraction of contact time within each risk level, for each activity.

3.4 Discussion

We already observed that in all summer camps there is a non-negligible percentage of contacts at high risk of contagion and that this is in general not due to some specific individuals or couples of individuals but the risk is quite homogeneously distributed among all the participants (Figures 3.4–3.5). We now discuss more in detail the results and their implications.

3.4.1 Social bubbles and roles

To analyze the effectiveness of the social bubble policies, we look at Figure 3.6, which reports the percentages of contacts taking place inter- and intra- groups and between children and children, educator and educator, and educator and children for the three summer camps. Note that in DAY-INT there was only a single group. We observe, as expected, that intra-group contacts are more numerous, but they are also interpreted as less risky since they are foreseen and permitted within the social bubble policies. On the other hand, inter-group contacts happen across different groups and are generally more risky; however, their limited number is a good indication of the effectiveness of the application of the social bubble policies. The collected data thus confirm that in case of an epidemic spreading in these settings, most of the possible contagions would likely be restricted to a single group, and transmission to other groups would be avoided or limited. Focusing on the interactions within each group, we observe that the highest percentages of contacts with high or medium-high risk of contagion involve children (i.e., children-children or educator-children CPIs), while the educators tend to have low-risk interactions among them.

3.4.2 Activity Type

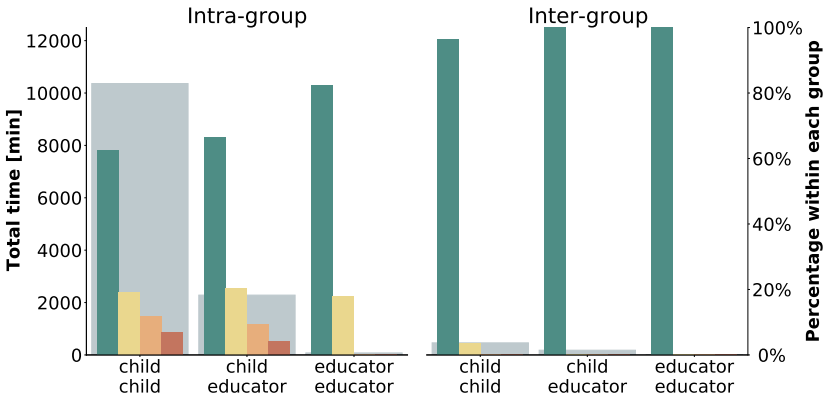
For summer camp AM-PRI shown in Figure 3.7a, it is evident that the activity involving the highest number of interactions per unit time is "snack"; however, it is also the only activity where none of the CPIs was at high risk. This is actually by design as the activity duration is less than 15 min (Table 3.2), which is the minimum duration required to mark a contact as high risk (Table 3.5). We observe a similar finding in the other two data sets, DAY-PRI and DAY-INT (Figures 3.7b and 3.7c), where "lunch" is the activity with the fewest risky contacts. This is probably because, during meal times, the children were not wearing their face masks; thus, the educators were paying more attention to the compliance to physical distancing rules. Moreover, the children were seated during lunch, so there was a reduced probability of accidental CPIs.

Other low-risk activities in AM-PRI were "crafts", "theater" and "team games", all meticulously organized activities where the educators established precise rules for physical distancing to avoid CPIs. The risk rises instead with "soccer" and "woods", where no precise rules were established, and the children were free to move in a large space. Moreover, these activities took place outdoor, and there is evidence for a reduced transmission risk during outdoor activities as compared to indoor ones [56, 57, 15, 90].

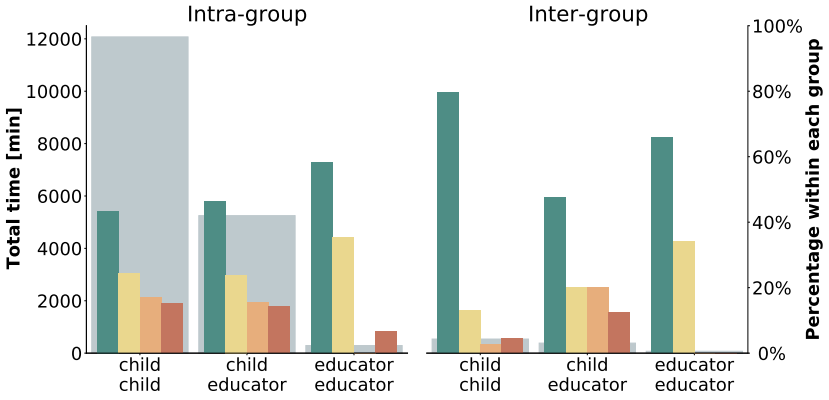
The riskiest activities, still with a limited total duration of high-risk close proximity contacts, are represented by "newspaper" and "board games", two indoor activities with specific constraints: the first consisted of collaborating in pairs in front of a computer, working on the summer camp's newspaper, and the second one consisted of playing board games around a table. Since the activities required being close to each other watching the same screen or table, the physical distance clearly could not be very large. However, it is worth highlighting that children wore face masks during the activities, thus reducing the transmission risks [62, 14, 86].

Moving to DAY-PRI, a different summer camp with a different organization (Figure 3.7b), we observe a high number of contacts during the activity "team games", even if most of these contacts are at low risk of contagion. Interestingly, in this summer camp the organized games imply many more contacts per unit time with respect to "free play". However, the activity with the highest percentage of high-risk CPIs is "day closing", which was the final part of the day, when children were waiting for pick up and entertained themselves by playing table tennis or table football, in rather unstructured way.

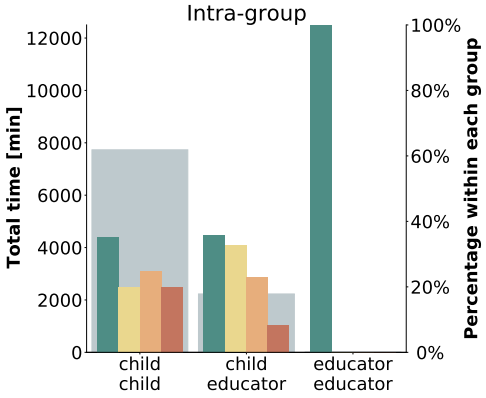
An additional and final scenario can be observed in DAY-INT, showing different typical behaviors, possibly due to a higher age range of the participants, namely



(A) AM-PRI

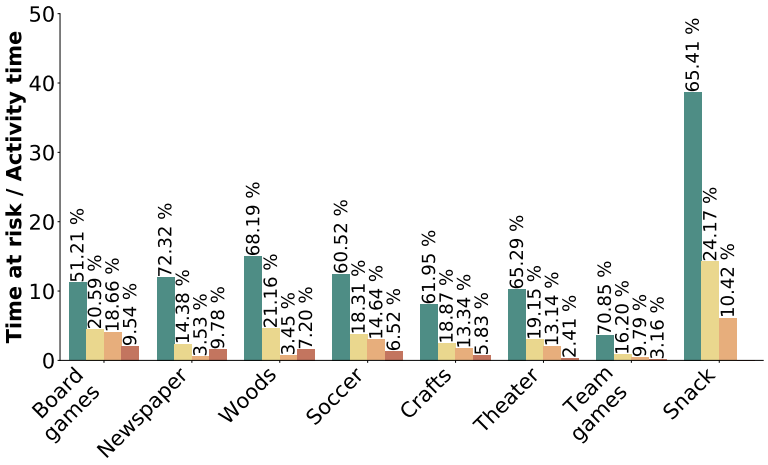


(B) DAY-PRI

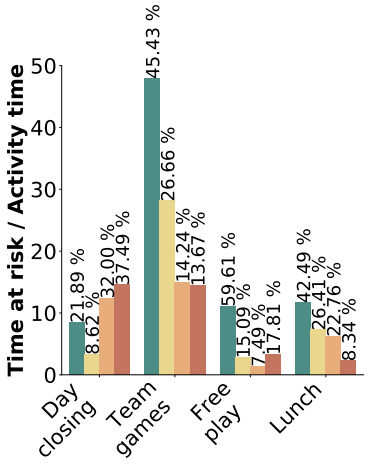


(C) DAY-INT

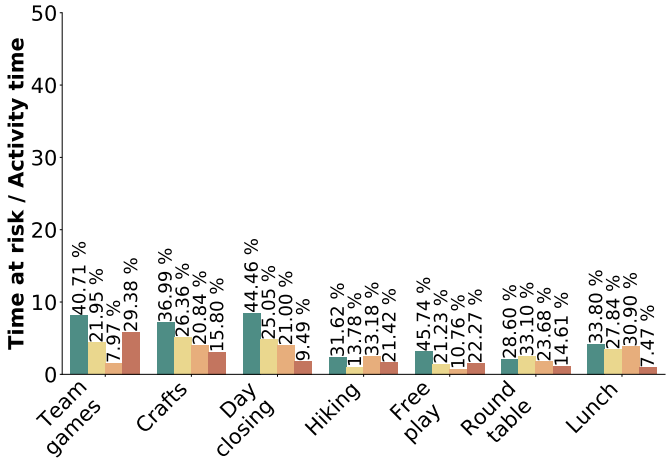
FIGURE 3.6: Social bubble policy and roles. Distribution of risk levels by group and type of interaction for each summer camp. The color bars, which refer to the right-hand scale, report the percentage of time of contact within each risk level. The grey background bars, which refer to the left-hand scale, report the total time of contact for each of the six categories.



(A) AM-PRI



(B) DAY-PRI



(C) DAY-INT

FIGURE 3.7: Activities and risk levels. The figure shows the distribution of the risk levels by activity, sorted according to a decreasing percentage of high-risk contacts for AM-PRI (Figure 3.7a), DAY-PRI age range 6-11 (Figure 3.7b) and DAY-INT 11-14 (Figure 3.7c). The percentages show the fraction of contact time within each risk level, for each activity.

11-14 years old, and different adherence to physical distancing rules. Figure 3.7c shows a general lowering of the time spent interacting with each other and, at the same time, a higher percentage of high-risk CPIs. Differently from AM-PRI but similarly to DAY-PRI, we observe that the activities with the highest risk are exactly the most organized ones: "team games" and "craft", followed by the ones where children were more free to move around: "day closing", "hiking", and "free play". The activities that provide less high-risk CPIs are instead "round table" and "lunch", where participants were sitting to talk or eat, all together but keeping a well-defined physical distance from one another.

All together, this analysis of the activities shows the different ways in which different settings have been addressed. In particular, it seems that the combination of mask-wearing in the close-interaction static activities and a precise organization of the dynamic activities results into an overall effective strategy to contain the risk.

3.4.3 Limitations

As with any experimental data collection, we acknowledge the limitations of our study. First, the gathered data sets are limited in time by the duration of the summer camps (one week, and half or whole days only) and by the number of participants (61 individuals in total). While the high temporal and spatial resolution enabled by Janus allow interesting analyses, the sample size and length limits make it impractical to simulate an epidemic spreading model based on this population. Further, all the summer camps were located in the Trentino area, and do not necessarily directly translate to other cities, regions, or countries, perhaps with different distancing rules.

Finally, a comparison to similar studies in the summer camp setting is not possible, as none are available in the literature. Moreover, we do not have hard ground truth to compare against; this would have required either cameras or manual annotations, which would have greatly interfered with the children privacy and the camps' activities. Nevertheless, the results and findings we outlined have been shared with the educators, who confirmed them based on their knowledge and recollection of the activity organization, and the observed general behavior of the children and educators.

Despite these limitations, we reassert that the data collected by the Janus devices is, to the best of our knowledge, the only example of physical distance data for child interactions with high spatio-temporal resolution collected during the COVID-19 pandemic.

3.5 Concluding Remarks

Tracking and measuring CPIs in a real setting is a challenging task that, however, plays a crucial role in understanding the dynamics of social interactions during the pandemic and their effect on the spread of the disease.

This work shows that the Janus system is well-suited to provide high temporal and spatial resolution data to capture CPIs in complex settings like summer camps.

In particular, we have analyzed three summer camps' daily activities and social interactions in the Autonomous Province of Trento (Italy). The captured CPIs allowed us to derive several key insights into the duration and proximity patterns characterizing the child-child and the educator-child interactions.

Specifically, we verified the effectiveness of the social bubble strategy, which is easy to implement in the summer camp setting and offers an effective mechanism to balance control of the epidemic against light restrictions on the children during educational and recreational experiences.

Moreover, we analyzed the risk levels of a series of activities performed during the summer camps. We obtained key information into their safety in terms of number of contacts, duration of the contacts, and level of contagion risk. When

combined with other metadata such as the location (indoor vs. outdoor) and the possibility to adopt personal protective equipment (i.e., face masks), this information can be exploited towards actionable policies to design safer environments for interactions among children in the summer camp setting but also at schools.

Chapter 4

Hermes: Low-Latency Proximity Detection

While Janus has been clearly shown to be a viable and flexible solution for proximity detection for a variety of applications, its separation of discovery and ranging across the two radios causes Janus to incur a non-negligible delay between these two elements of proximity detection. Put another way, while Janus is configured to know the existence of another node within the discovery latency epoch, the distance to this node will not be known until the coordination process between the BLE and UWB radios concludes, a process that takes between two and three epochs.

In this chapter, we explore an option that allows triggering ranging immediately after discovery. In brief, we remove the BLE radio, performing all operations over UWB. We return to the original BLEnd neighbor discovery protocol, exploring the implications of switching from BLE to UWB and the opportunities to extend the protocol to naturally incorporate ranging.

One of the primary benefits of using BLE for continuous neighbor discovery in Janus is its energy efficiency. Therefore, in the development of Hermes[1] we pay close attention to the energy trade-offs associated with our choices.

4.1 Hermes In a Nutshell

The core of Hermes is a continuous neighbor discovery protocol using UWB instead of BLE and thus able to exploit the transmitted messages both for discovery and as the initiation of two way ranging. On the surface, the protocol takes inspiration from U-BLEnd [53]: w.r.t. F-BLEnd, used in Chapter 1, in U-BLEnd the node is active only in the first half of the epoch, and idle for the other half to save energy. Hermes, as seen in Figure 4.1 schedules two key elements: a `train` and a series of scans. The `train` itself is a sequence of `trainslots`, each composed of a transmission and a scan. The transmission serves both as a beacon that advertises the presence of the transmitter, and, if received, it also serves as the first poll message in a TWR operation.

The series of scans are separated by no more than the length of the `train`, reduced by the length of one train slot to ensure that at least one transmission is detected. If a scan does not receive a message, the node returns to a sleep state to save energy. Instead, if it receives the beacon of another node, it replies with the TWR response, then listens for the response in the subsequent `trainslot`, enabling its own calculation of the distance between the two nodes.

Certain elements of Hermes are fixed while others depend on the scenario. The length of a single `trainslot` is lower bound by the choice of UWB technology (see details in Section §4.3.1). The time between scan slots must be less than the length of the `train`. The number of scheduled scan slots must be such that the

[1] The protocol is named after the Greek messenger god, Hermes, known for his speed, reflecting the low latency goal of the protocol.

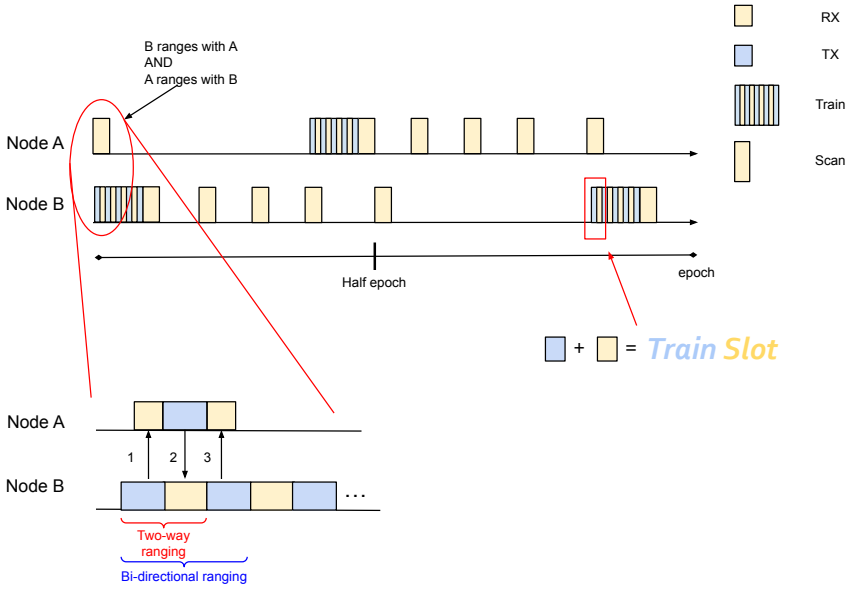


FIGURE 4.1: Hermes protocol overview. Details of a single discovery/ranging are shown.

last scan falls just after the halfway mark of the epoch:

$$\text{numberScans} = \text{ceiling}(\text{epochLength}/2/(\text{trainDuration} - \text{trainslotLength})) \quad (4.1)$$

The first parameter we consider in depth is the epoch length itself, or the period at which the node repeats the discovery and ranging procedure. Short epochs allow the protocol to be more reactive, as discovery is guaranteed within the epoch, as in Janus. Nevertheless, small epochs utilize fewer scans, increasing the probability of collisions and thus reducing protocol reliability. Longer epochs, instead, include more scan operations, decreasing collisions and increasing reliability, but they target logging-style applications.

It should be stated that in the presence of message collisions, discovery is not *guaranteed*. In fact, the impact of collisions on the discovery rate is one of the primary points of focus for this exploration.

Epoch Length (s)	No. Trainslots	No. Scan slots	Epoch Length (s)	No. Trainslots	No. Scan slots
1	10	43	15	10	642
1	50	8	15	50	118
1	70	6	15	70	84
1	100	4	15	100	59
2	10	86	30	10	1283
2	50	16	30	50	236
2	70	12	30	70	168
2	100	8	30	100	117

TABLE 4.1: Sample parameter configurations for Hermes.

The number of `trainslots` in the `train` is also variable. Fewer `trainslots` consume less energy as part of the `train`, however, they require more scans to be scheduled. More `trainslots` increase the length of the `train` and correspondingly its consumption, but they also decrease the number of required scan slots. Table 4.1 offers some sample configurations. Given that the UWB radio will be required to ramp up and down before and after each scan slot, we can immediately see that some of these configurations are practically unrealistic.

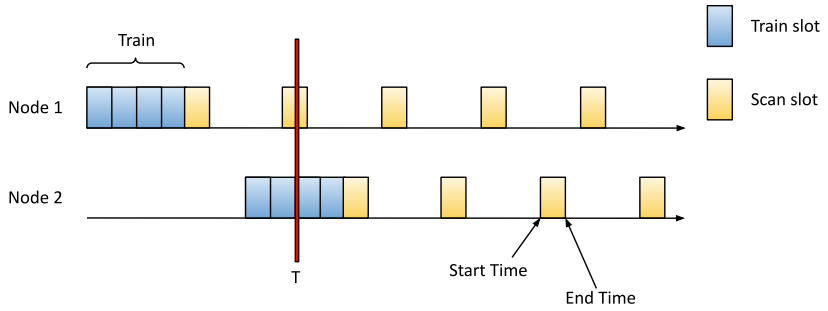


FIGURE 4.2: Illustration of Hermes in simulation showing two devices. The red bar shows an instant of time T in which one `trainslot` and one scan slot are simultaneously active, leading to discovery.

4.2 Simulation

To study the protocol operation, we designed and built a custom discrete event simulator in Python. It focuses on the discovery element of Hermes, considering the overlap of the `trainslot` transmission with a scan slot, identifying successful discovery as well as collisions that inhibit discovery. The simulator allows rapid study of the protocol parameters in scenarios with different node densities.

In addition to rapid protocol evaluation, the simulator also serves to validate the mathematical model and as a point of comparison for experiments.

4.2.1 Simulator Description

The simulator is a discrete-event simulator, where the outputs are triggered with the geometric overlapping of scans slots and `trainslots`. The possible events identified by the simulator are *discoveries* and *collisions* of messages exchanged among the *nodes*. In the simulator, device behavior is described within a Python class *node* and a Python generator class uses iterations to schedule `trainslots` and scan slots according to the provided protocol parameters. As the focus of the simulator is on discoveries between devices, the `trainslot` and scan slot are monolithic *slots*, with no distinction between transmission and reception. In other words, the simulator has no definition of the radio channel, but the *shared medium* is represented by the *active slots*, either `trainslot` or scan slot, present at a certain moment. Each slot has a *start time* and *end time*, identifying when they are active on the shared medium.

Figure 4.2 offers a graphical representation the protocol with two devices. The `train`, as defined in §4.1, is composed of `trainslots`, each of which is represented in the simulator as a single slot, not divided into transmission and reception. Note that the simulator does not simulate the TWR exchange of Hermes. Each slot, has a start time and end time allowing the simulator to identify when two (or more) slots overlap, i.e., are active at the same instant in time. For instance, at time *T* shown in the figure, the scan slot of Node 1 and the `trainslot` of Node 2 are active at the same time, leading to a *discovery* between the two devices: Node 1 discovers Node 2 and vice versa.

In the case when more than two nodes have a slot active in the same instant in time, the simulator must consider how to handle the situation, specifically regarding message collisions. Our choices are summarized in Figure 4.3, which shows on the left the standard overlap resulting in discovery when a `trainslot` overlaps with a scan. When two or more `trainslots` overlap with the same scan slot and their start times are greater than the start time of the scan (the beginning of the `trainslots` are detected), we consider this *shadowing*. In this case, Node 1 discovers Node 2 and vice versa, while Node 3 is *shadowed* by Node 2 and is not

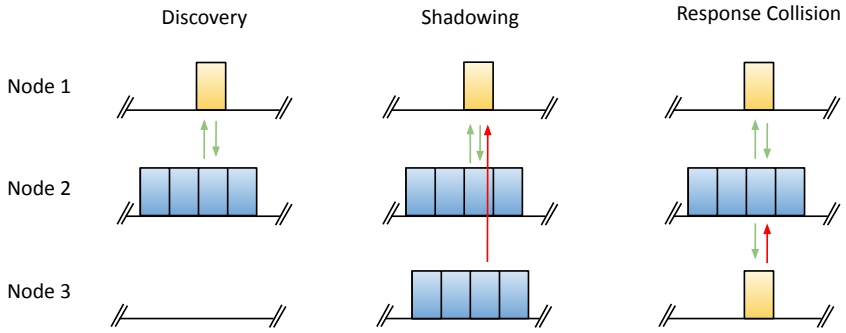


FIGURE 4.3: Concurrent interactions of slots for more than two nodes. Green arrows represent successful message exchanges, while the red arrows depict a failed exchange.

discovered by Node 1. The intuition is that when two or more devices transmit at the same time, i.e., when two ore more `trainslots` overlap, thanks to the UWB PHY layer they will not completely collide and instead one of them will be correctly received. Specifically, the first detected preamble is correctly received, while the other is discarded.

A *response collision*, shown on the right of Figure 4.3, occurs when multiple scan slots overlap with a single `trainslot` and the `trainslot` has a start time greater than the scans. In this case, both Node 1 and Node 3 discover Node 2, since they both receive its transmission. However, since the replies are sent at the same time, only one among them is captured by Node 2. Therefore, Node 1 and Node 3 discover Node 2, while Node 2 discovers only Node 1.

Inside the simulator, we track the following:

- *Discovery*: number of nodes discovered per epoch, either with the `train` or with the scan. In Figure 4.3 on the left, Node 1 adds Node 2 to its discoveries and vice versa.
- *Missed Trainslots*: number of nodes missed per epoch due to shadowing. In the middle of Figure 4.3, Node 1 adds Node 3 to its missed `trainslots`.
- *Missed Scans*: number of nodes missed per epoch due to response collision. In the right of Figure 4.3, Node 2 adds Node 3 to its missed scans.
- *Discovery WITH train*: number of nodes discovered per epoch during the `train` phase. In the right of Figure 4.3, Node 2 discovers Node 1 during the `train` of Node 2.
- *Discover WITH scan*: number of nodes discovered per epoch during the scan phase. In the left of Figure 4.3, Node 1 discovers Node 2 during the scan of Node 1.

These five simulator outputs are used to evaluate the protocol in the following.

In addition to the basic protocol implementation described in §4.1, the simulator and the practical implementation include a random *jitter* at the beginning of each epoch. This feature is shown in Figure 4.4 and is fundamental to avoid continuous collisions between overlapping trains. In fact without jitter, if two `trainslots` or two scan slots from two different node overlap, they will continuously do so across all epochs. The jitter is a uniformly distributed variable from λ_1 to 0, with $\lambda_1 < 0$. We use a negative jitter, shifting the start of the epoch earlier to prevent a "void" before the epoch starts: the presence of a scan slot immediately after the epoch half guarantees that, without collisions, there is always discovery between two devices. A positive jitter, instead, would shift the beginning of the subsequent epoch forward, "breaking" this important assumption and decreasing the discovery rate. The impact of the jitter is evaluated in Section §4.2.2.

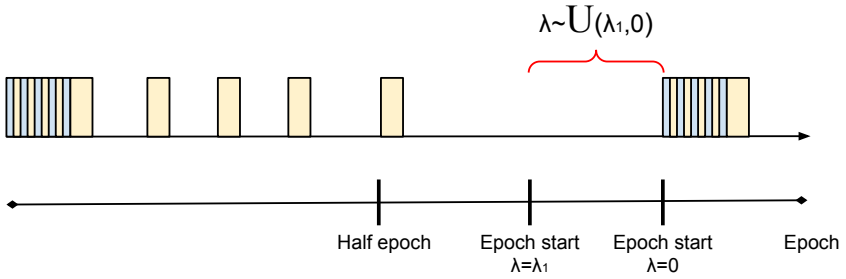


FIGURE 4.4: Hermes jitter. λ is the jitter, and is a uniformly distributed random variable from λ_1 to 0, where λ_1 is a negative value.

4.2.2 Simulation Results

As stated previously, the simulator provides a valuable tool for understanding Hermes. We focus most of our exploration on the discovery rate, first using the simulator to understand the impediments to discovery, then pushing the protocol in a variety of settings. We then use the simulator to explore in detail the impact of jitter and its effect on latency.

In all simulations, unless otherwise specified, each node starts at a random time within the first epoch. The simulation runs for 10 epochs and all simulations are repeated 1000 times. We set the maximum jitter equal to the `train` length.

Understanding Discovery Rate

We begin our evaluation with a scenario with a 2 s epoch and 10 neighboring devices. Before looking at the Hermes results, we remind the reader that in the original Janus protocol, one key parameter of neighbor discovery is the *minimum discovery probability*, typically set at 95%. This takes into account collisions that prevent discovery within the target latency. Notably, due to the jitter added to the protocol, most of these missed discovery events are recovered in later epochs, allowing Janus to eventually discovery and range with all neighbors. Hermes does not explicitly consider the minimum discovery probability, but collisions do negatively affect the discovery rate, as described previously with shadowing and response collisions.

Figure 4.5 shows in yellow the overall system discovery rate as we vary the number of `trainslots`. We immediately see a knee in the plot, indicating that the highest discovery rate of 90.65% is achieved with 20 `trainslots`.

To understand this, we recall that each node will discover on average half of the nodes during its `train`. The other half will be discovered during scan slots. With only 5 slots in the `train`, nearly every one of these slots must be used to successfully discovery one of the other 9 nodes in the system. Due to the random nature of the protocol, it is very likely that response collisions will occur, reducing the discovery rate. Indeed, the blue plot of Figure 4.5 shows that with 5 `trainslots`, approximately 30% of the scan responses collide. Instead, as the number of `trainslots` increases, the probability of response collisions decreases, shown as the decrease in the blue line, and the discovery rate increases, shown as the increase in the yellow line.

However, as the `train` length increases, the number of scan slots decreases as the interval between them increases. Therefore, each scan has a higher probability that it will be required to hear more than one of `trainslots`- a situation that results in Shadowing of one of the nodes during its `train`.

The knee of the discovery rate curve occurs when the combined impact of Shadowing and Response Collisions is minimized.

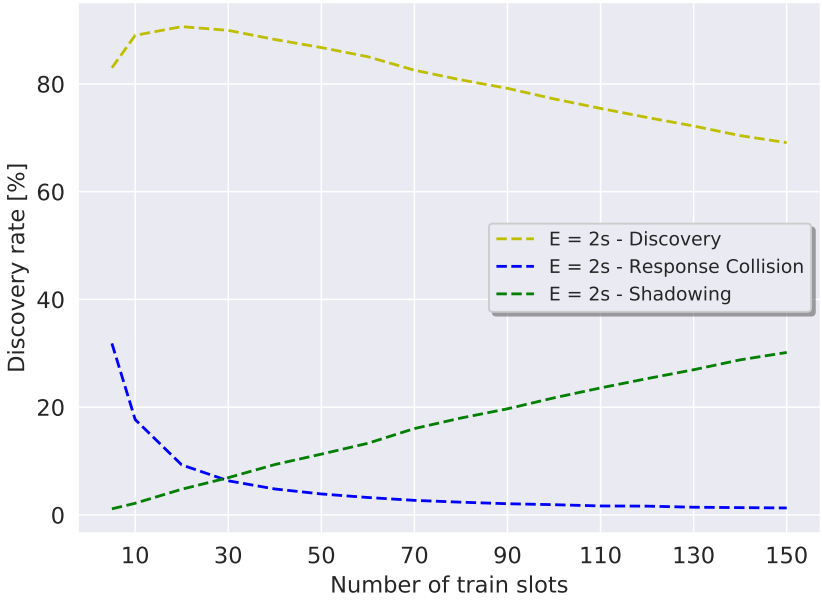


FIGURE 4.5: Simulations with 10 nodes and 2 s epoch length to understand the impact of collisions (Shadowing and Response collisions on the discovery rate as the `train` length increases).

Evaluating Discovery Rates

We next exploit our simulator to explore a wide range of Hermes parameters. In all cases, we vary the number of `trainslots`, as in the previous section, examining the principal objective of the protocol, discovery rate, for varying epochs. Unlike with Janus, where the epoch length typically corresponds to the discovery latency, in Hermes, the epoch length corresponds to *both* discovery and ranging latency. We also show the behavior with differing numbers of neighbors, as this affects the number of messages on the shared UWB communication channel, and thus the possibility for collisions.

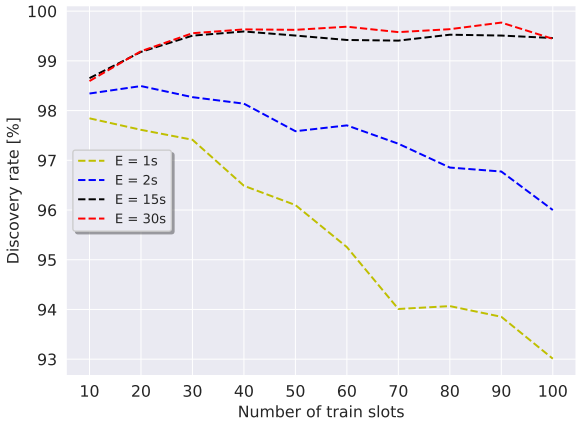
Examining simulation results across the plots in Figure 4.6, we first notice that as we increase the number of nodes, the discovery rate decreases. This is due to the increased message collisions. While this implies that Hermes performs *poorly* for crowded settings, scenarios for proximity warning typically have few nodes and are expected to perform well.

We also see that discovery rates are quite high for long epochs. As noted in Section §4.2.2, once the `train` is long enough to accommodate half of the discoveries without collision, the long epoch results in the placement of relatively many scans, which are therefore unlikely to result in Response Collisions.

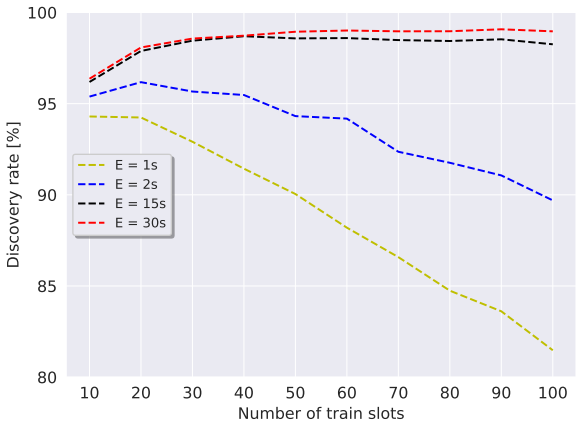
Evaluating Energy Trade-off

While our goal with Hermes is to have a low latency protocol with a high discovery rate, we need to balance this with the energy cost of the protocol configuration. Figure 4.7 models the lifetime of a Hermes node performing discovery-only. As with the other curves, we clearly see a knee in the curve, showing that as the `train` length increases the consumption increases to a point, then begins to decrease. This is due to the relationship between the length of the `train` and the spacing between the scan slots. As the `train` increases, the number of scans required decreases.

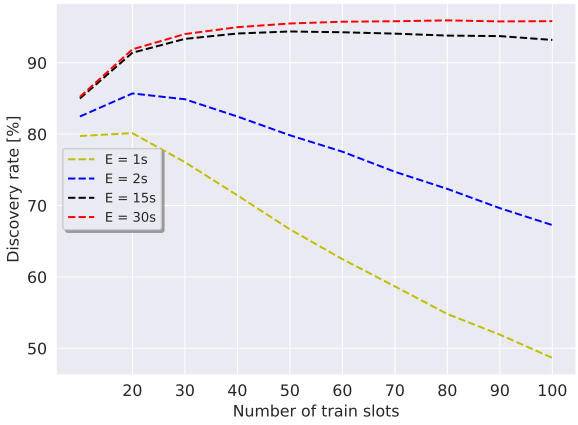
Notably, the point that minimizes the consumption does not correspond to the point in the previous figures with the highest reliability. This discrepancy forces application designers to consider the trade-offs as appropriate for their application, e.g., sacrificing lifetime to improve discovery rates, or vice versa, accepting a lower



(A) 3 nodes



(B) 5 nodes



(C) 15 nodes

FIGURE 4.6: Discovery rates VS. `train` lengths for varying epoch lengths.

discovery rate to save lifetime. The simulator offers a valuable tool to explore these trade-offs.

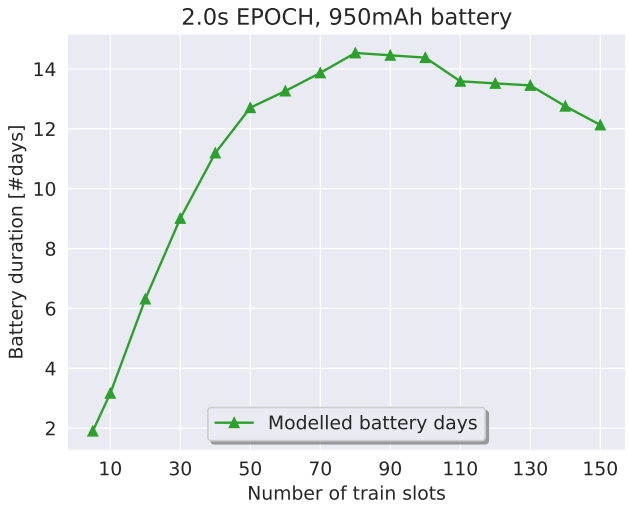


FIGURE 4.7: Lifetime of Hermes vs `train` length for a 2 s epoch.

Evaluating Discovery across Multiple Epochs

Due to the formulation of the Hermes protocol, without communication errors such as those arising from collision, all nodes will either directly discover one another or be discovered by another node within one epoch. In practice, collisions reduce the discovery rates, as seen in the previous sections.

To combat these errors, Hermes incorporates an intentional jitter, to allow the communications among pairs of nodes to shift w.r.t. one another across epochs. The driving motivation is to avoid communications that constantly collide across epochs due to unfortunate alignment.

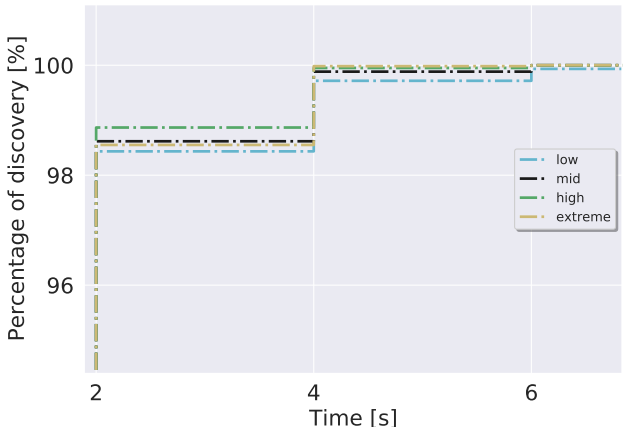
In this section, we analyze the positive effect of jitter, considering the effect on the discovery rate across multiple epochs. As our goal is to discover and range with all neighboring nodes, the ability of the protocol to reach this goal within multiple epochs is of great, practical importance.

Figure 4.8 shows the discovery rate for 3, 5 and 15 nodes. In all cases, we configured a 2 s epoch with 20 `trainslots`, a configuration that maximizes the discovery rate in a single epoch. Within each plot, we show four different configurations for the jitter:

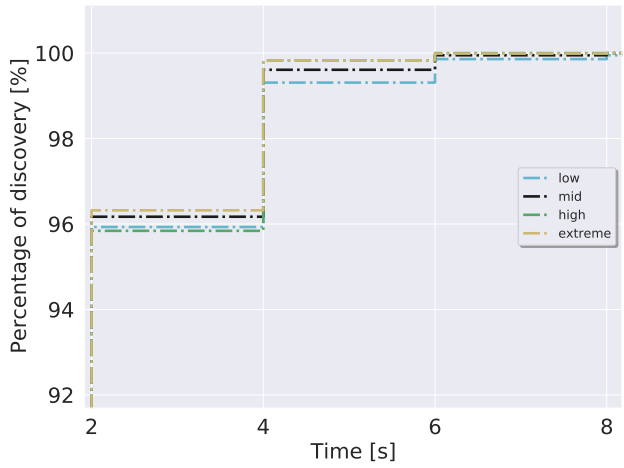
- Low: maximum jitter is equal to the `train` duration
- Medium: maximum jitter is 5 times the `train` duration
- High: maximum jitter is 10 times the `train` duration
- Extreme: maximum jitter is 20 times the `train` duration

In all figures, we see the CDF of the nodes discovered *per epoch*, hence the stepwise nature of the plots. After the first epoch, the curve with the lowest jitter meets the discovery rate shown in Section §4.2.2. Changing the jitter affects the discovery probability. Larger jitter notably allows the trains to shift more within the epoch, therefore, if the trains were colliding in one epoch, they will be less likely to overlap in subsequent epochs. Instead, keeping the jitter below the `train` length performs the worst, resulting in the lowest discovery probability.

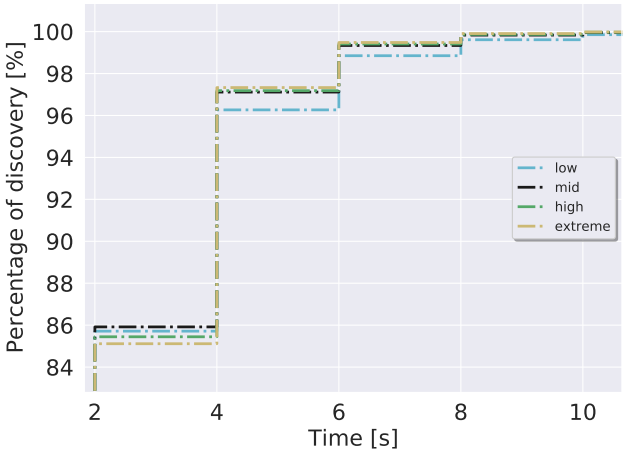
Notably, however, Hermes achieves more than 96% discovery within two epochs, even in the most challenging scenario with 15 neighbors. This value should be taken in comparison to the discovery rates of Janus, which were guaranteed to be up to 95% within the defined latency, and ranging would be scheduled up to two epochs later. Instead, in Hermes, ranging occurs at the same moment as discovery, therefore 96% or more of the ranging will occur withing the first two epochs.



(A) 3 nodes



(B) 5 nodes



(C) 15 nodes

FIGURE 4.8: Discovery rates VS. Latency for varying jitter.

4.3 Implementation

Thanks to the positive results achieved through simulation we decided to shift our focus to the real implementation of Hermes. We target the DWM1001 board from DecaWave, now QORVO, as done for Janus, a choice that allows us to directly compare the two protocols on the same grounds. While with the Janus implementation, one of the primary challenges was synchronization between the BLE and the UWB radios, in Hermes the precise timings and the energy consumption represent the central challenges. In fact, to limit consumption of the UWB radio, the transmission and reception timings must be reduced as much as possible, but without disrupting the correct functionality of the implementation. Nevertheless, even with the timings set correctly, consumption can still be unacceptably high, forcing us to explore other reduction techniques such as deep sleep and preamble sniffing.

This section describes our implementation of Hermes focusing on the UWB configuration details. We then move on to showing results, comparing them to simulation.

4.3.1 Hermes Configuration Details

The goal behind our choices for the transmission and reception times in Hermes was to minimize consumption and packet on-air time.

First, we choose the minimum possible UWB preamble guaranteed by DecaWave, 64 symbols (around 64 us), resulting in a TX slot of 225 us. Therefore, the `train` is composed of alternating TX of 225 us and RX of around 40 us. This RX duration is sufficient to detect UWB preamble symbols, verified experimentally. Between each TX and the subsequent RX in the `train` there is a fixed delay, a fundamental value for the TWR distance calculation. In our case this value depends on the type of message transmitted. For the `train` TX, this delay is 537 us, and represents the interval between the end of the transmission of the RMARKER and the start of the RX phase. Therefore, the interval between the RX and the consecutive TX must be greater, accounting for the time needed to complete the packet reception, around 113 us. This represents the minimum packet duration possible, since we used the preamble of 64 symbols and 6.81 Mbps data rate, the highest available. Thus, between two consecutive TXs, there are 1300 us, fixing the scan time to the same amount. This choice is mandatory to ensure that at least one TX among the two is detected by the scan.

4.3.2 Energy Saving Techniques: Deep Sleep and Preamble Sniffing

The use of techniques to save energy in a UWB-only protocol is vital. As shown in the QORVO documentation [82], the current profiles of both transmission and reception can reach peak values one magnitude higher than BLE.

The deep sleep mode of the microcontroller offers a well-known mechanism to reduce consumption. In Hermes, the device enters the deep sleep mode twice during the epoch schedule:

- *Second half of the epoch*: after the last scan slot, completed after the end of the first half of the epoch, the microcontroller goes to sleep. The wake up occurs 8 ms before the beginning of the subsequent epoch.
- *Between the scans*: between each scan slot, the device goes to sleep, with the wake up 8 ms before the subsequent scan.

The time needed for the radio to re-start is approximately 5.5 ms (§2.4), but with Hermes implementation any delay can lead to incorrect protocol operation. Therefore, we added a 2.5 ms guard time. This value was calculated empirically, and we hope to reduce it in future protocol iterations.

Deep sleep allows drastic reductions in protocol consumption. For instance, with 20 `trainslots` and 2 s epoch, the mean current consumed per epoch is *i)* 17.7305 mA without deep sleep *ii)* 6.2497 mA with deep sleep concretely showing the importance of this feature.

On the other hand, *preamble sniffing*, described in the DW1000 user manual [83], permits changes in the behavior of the radio listening state (RX state). The standard reception state of the DW1000 UWB radio is called *preamble hunting*: the radio is on, waiting to "capture" any UWB preambles. To do so, the radio remains active for the entire reception duration, forming a monolithic scan phase. Instead, using preamble sniffing, the on state of the radio is alternated with off periods during which the radio enters the IDLE state. In IDLE, the radio consumes approximately 12 mA and does not require any wake-up time to return to RX state. However, the introduction of off periods during the listening can affect the discovery rate, due to the "holes" inserted in the scan. The trade-off between the energy saved and the decrease in reliability is explored in Section §4.3.3, where we show the results of tabletop experiments.

4.3.3 Experimental Results

To conclude the discussion of Hermes, we show results with the real implementation. In the following, we first illustrate the reliability performance of the protocol, together with a comparison with the simulation results. Next, the energy consumption of the protocol is evaluated, using the same experimental configuration as used with Janus (§2.6.5). Finally, we activate preamble sniffing, showing its effect on reliability and energy consumption

Reliability

The goal of these experiments was to explore the discovery rate and compare it with the simulator. We show the experiments carried out with 3, 5 and 15 devices, evaluating the discovery rate against the number of `trainslots`. Considering the practical limitations of experimentation vs. simulation, we fixed the epoch length to 2 s to test the reliability with a *reactive* configuration, the typical application targeted by this protocol.

All experiments were conducted in the CLOVES testbed [69], a UWB network infrastructure developed at the University of Trento. The testbed consists of 275 UWB devices, installed below the ceiling tiles of the university buildings. Figure 4.9 shows the experiment area with 7 ceiling-nodes and 8 "fly-nodes": additional nodes placed on the floor and raised on wooden sticks, extending number of nodes in the restricted experimental area to 15. The experiments with 3 and 5 devices are conducted using only devices attached to the ceiling tiles. For each `train` length we repeated the experiment 12 times, recording 40 epochs each time for a total of 16 minutes per configuration.

We begin our analysis in Figure 4.10, showing experiments with 3 devices. As shown previously in simulation, with a 2 s epoch and 3 nodes (Figure 4.6a), for the `train` lengths considered, the discovery rate ranges from 95% to 98%. This range is confirmed in experimentation, with the discovery rate always above 95%, with the peak at 97.5%. Interestingly, the trend from 10 to 30 `trainslots`, follows that seen in simulation, with a monotonically increasing discovery rate. However, higher discovery rates occur around 30 and 40 `trainslots`, rather than at 20 slots seen in the simulator. In experimentation, a clear peak is not visible, instead a plateau of "higher rates" is seen between 30 and 40 `trainslots`. It should be noted that the drop in the discovery rate from 10 to 100 `trainslots` is only 2-3 percentage points in the simulator, making it difficult to demonstrate the change in the rate with real experiments. Further, the blue lines show the minimum and maximum values of the 12 experiments executed per `train` length. The high variance indicates that more experiments should be carried out to refine the results, nevertheless the trends are clear.

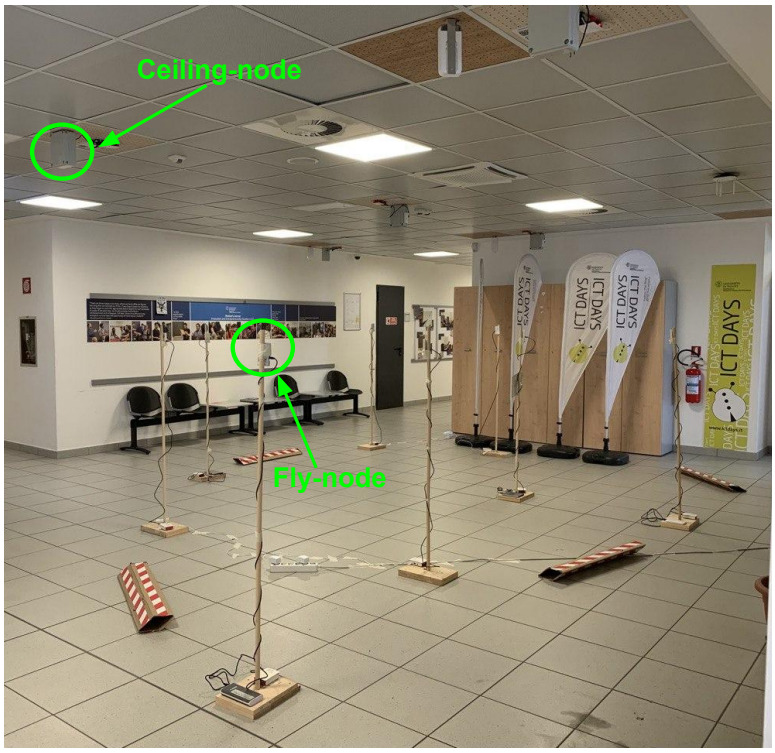


FIGURE 4.9: Photo of the experiment area in the CLOVES [69] testbed. 7 devices are installed on the ceiling and 8 temporary, *fly-nodes* are placed with wood sticks.

The trend in comparison to simulation is also confirmed in Figure 4.11 where results with 5 nodes are depicted. Indeed, in the first part the rate increases until it reaches a plateau. However, the plot also shows a shift of the curve with the "peak" around 40 `trainslot` with a 94.8% discovery rate, and the plateau ranging from 40 to 70 `trainslots`. In the simulator, instead, the rates are 2% higher, and the plateau occurs between 10 and 40 `trainslots`. These discrepancies can be considered acceptable considering the variance of the experimental data and the fact that the the simulator is an estimate and does not include a realistic channel model.

This is also evident in Figure 4.12 where the discovery rate with 15 devices is shown. Notably, the trend here is the same as in the simulator, with a clear decrease for more than 30 `trainslots`. These results illustrate on the one hand that the predictions made by the simulator are of value even with more devices, while on the other hand, the drop in reliability w.r.t. the simulator arising from collisions is notable. Indeed, if the simulator with 30 `trainslots` achieves a discovery rate of 85%, the corresponding value in testbed experiments is 75%. We presume that the greater number of transmissions produces a greater number of disruptive collisions, which are not accounted for in the simulator at this time.

Overall, the simulator correctly predicts the reliability of the protocol, especially with a low number of devices. The high variance of the real experiments further impedes the reproduction of the simulator curves, suggesting that a more intensive experimental campaign is needed to achieve a more precise results. However, the trend of the plots is already visible, especially with 15 devices, demonstrating that the simulator is in line with the protocol implementation.

Energy Consumption

The current consumption of Hermes has been evaluated using a Keithley SourceMeter 2450 and the Janus custom tag described in Chapter 2, Section §2.5. Regarding reliability, we fixed the epoch length to 2 s and used two `trainslot` lengths: one

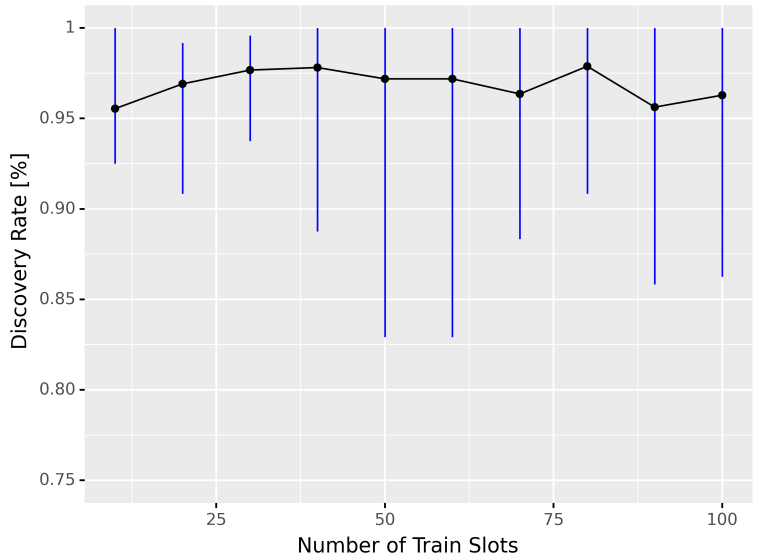


FIGURE 4.10: Discovery rates VS. `train` length with 3 nodes.

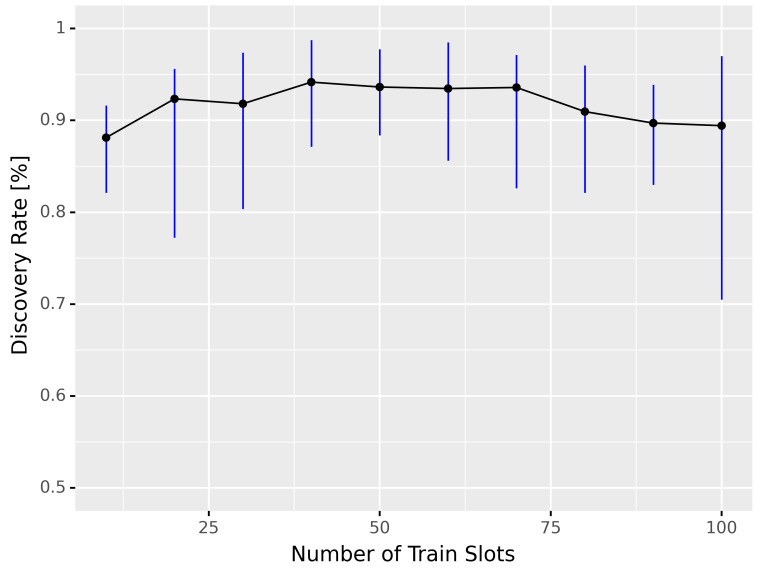


FIGURE 4.11: Discovery rates VS. `train` length with 5 nodes.

maximizing the discovery rate according to the simulator and one minimizing consumption according to the energy model shown previously (4.7). The results are shown in Table 4.2. For each `trainslot` we tested the custom tag alone, with 4 neighbors and with 9 neighbors. The neighboring devices were at most 2 m from the tag connected to the SourceMeter. To allow the connection between the SourceMeter and the costum tag, these experiments have been conducted in our lab, with the devices on the desks.

While 20 `trainslots` is one of the values that maximize the discovery rate, the current draw of Hermes is quite high, above 6.6 mA. This is particularly high when seen in comparison to an isolated Janus device, which, thanks to the low-power BLE radio, consumes 1.1 mA in the reactive configuration, i.e., 2 s epoch length. The situation improves with 80 `trainslots`, where the consumption falls to 3.2183 mA, which is still higher than Janus. Note, however, that the reactive scenario in Hermes not only achieves discovery, but also ranging within the 2 s time limit. Thus Hermes is approximately 2-3 times more *reactive* than Janus. We

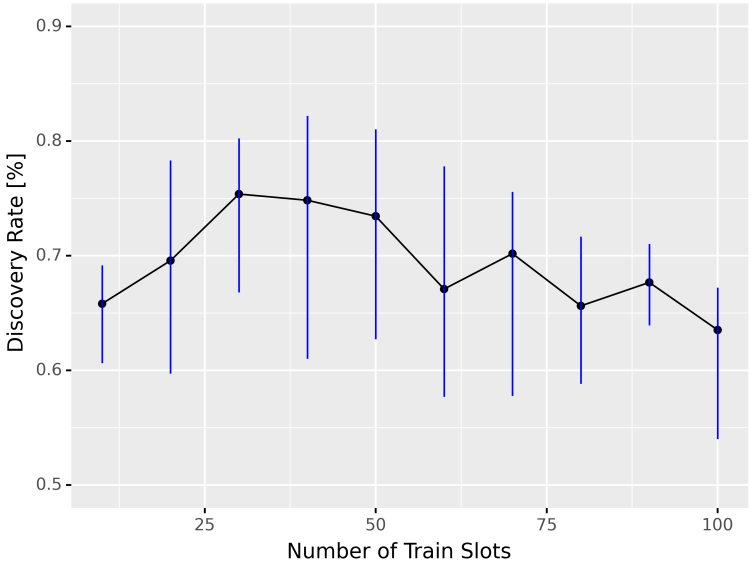


FIGURE 4.12: Discovery rates VS. `train` length with 15 nodes.

TABLE 4.2: Current consumption of Hermes protocol.

	TS = 20 (Max Discovery Rate)	TS = 80 (Min Consumption)
Alone	6.2497 mA	3.2183 mA
4 Neighbors	6.1909 mA	3.1903 mA
9 Neighbors	6.1041 mA	3.1453 mA

also recall that in Hermes, we fully expected to increase consumption w.r.t. Janus, thus higher consumption values are acceptable. Nevertheless, we note that while the consumption with 80 `trainslots` seems a reasonable value, the discovery rate at this `train` length drops to 92% with 5 nodes, a value only slightly below the 95% discovery probability established for Janus.

Interestingly, while for Janus consumption increases with an increasing number of neighbors due to the addition of UWB slots for ranging, in Hermes we actually see that the consumption decreases with an increase of neighbors. While this behavior is counter intuitive, it is likely due to the difference in scan slot management when neighbors are present or not. Specifically, without preamble sniffing, a scan keeps the UWB radio active for 1.3 ms in RX mode. Considering that TX has lower consumption than RX, when a scan detects another node and starts to participate in TWR, the RX is interrupted and instead the actual reception is followed by a TX and finally by a brief RX. This results in a scan phase with shorter active duration, reducing consumption. Indeed, this effect increases with the increase of neighbors, with more scan slots triggering discovery and thus ranging.

Preamble Sniffing Evaluation

To evaluate the effectiveness of preamble sniffing, energy consumption and discovery rate must be considered as the introduction of "off periods" during the scan can negatively impact discovery rate. In Table 4.3 we explore this trade-off, with multiple preamble sniffing configurations. In this case, the experiments have been conducted in the testbed, considering only two devices. As before we fixed the epoch length to 2 s and the number of `trainslots` to 20.

In Table 4.3, T_{on} represents the duration of the "on" time of the radio in scan mode in microseconds (μs). T_{off} is the "off" duration. The duty cycle, DC, is given

TABLE 4.3: Preamble Sniffing evaluation: considering discovery rate and current consumption for various duty cycle configurations.

T_{on} (μs)	T_{off} (μs)	DC (%)	Discovery rate (%)	Consumption (mA)	Relative consumption (%)
-	-	100	100	6.2497	100
48	32	60	93.3	5.6588	90.5451
32	32	50	90.83	5.2689	84.3064
16	32	33.33	87.91	4.8971	78.3574

by the ratio between T_{on} and the sum of T_{on} and T_{off}. We tested four DC values, using a configuration with preamble sniffing disabled as the baseline.

We begin our analysis by setting the DC to 60%. The amount of energy saved is shown in the last column, where "Relative consumption" represents the ratio between the current consumed with and without preamble sniffing. While we save approximately 10% in energy consumption, we lose approximately 6% of the discovery rate. We note that even if the duty cycle is reduced by 40%, we *only* save 10% overall. This is due to the fact that only the scan elements of the protocol are subject to the savings due to preamble sniffing. Further, during the off periods, the radio switches to the IDLE state, with a current draw of 12 mA, not zero.

The situation is slightly worse at a duty cycle of 50%, with the discovery rate dropping to almost 91%. However, the consumption decreases more than the discovery rate, with energy savings of 16%. Finally, with a duty cycle of 33.33%, consumption drops significantly, saving almost 1.5 mA. However, we also see a significant drop in discovery rate, with a value well below 90%. This could be unacceptable, especially for applications such as proximity warning system in which the reliability is crucial. Overall, a T_{off} of 32 μs, although it may allow for energy savings, creates a non-negligible reliability loss. Indeed, being equal to half the duration of the considered preamble ($\sim$ 64 μs), the gap that is created between two listenings could potentially risk losing the correct reception of the preamble. Future analysis will focus on the test of different configurations, with lower values of T_{off}.

4.3.4 An Eye to the Future: From the DW1000 to the DW3000

Since its release around 2010, the DW1000 module by Decawave (now QORVO) [82], has contributed significantly to the design and development of multiple IoT applications based on UWB. Nevertheless, it does present several limitations, which are inherited by all results in this thesis. Importantly, however, the results presented here can also benefit from hardware improvements.

For example, in 2020, the new QORVO DW3000 UWB module [85] entered in the market, promising to drastically enhance performance w.r.t. the DW1000. A recent comparison [80] between the two modules considers both ranging quality and energy consumption. Through experimentation, the authors demonstrate a 30% reduction in energy consumption from the DW1000 to the DW3000 during ranging. Specifically, the radio RX phase, the highest consuming element of ranging, decreases from $\sim$ 240 mW to $\sim$ 120 mW.

By shifting the protocols of this thesis to the new module, we can directly benefit from these improvements. Notably, for Janus, the "high consumption" of the DW1000 is less relevant due to its use of the BLE radio for continuous ranging. Instead, Hermes should see larger improvements as all operations use the UWB radio. In general, shifting to the DW3000 radio would allow us to revisit the design trade-offs, e.g., the use of preamble sniffing to save energy, but at the price of a loss in reliability.

4.4 Concluding Remarks

In this chapter we introduced Hermes, a low-latency proximity detection protocol built atop only the UWB radio. Hermes fuses continuous neighbor discovery with ranging, drastically reducing the latency between the two operations in the Janus protocol. This merge incurs a reasonable cost in energy consumption and produces acceptable reliability.

We showed through simulation that the protocol can achieve good performance and used this tool to provide insights on the reliability and the latency of Hermes in different configurations. With the simulator we were able to show that the protocol can easily guarantee discovery rates above 95% with increasing numbers of neighbors. Moreover, we have shown that, thanks to the low-latency achieved by the protocol, more than 98% of the neighbors are discovered within two epochs with the possibility to achieve ranging at the same time.

Finally, we implemented the protocol for the DWM1000 UWB radio, and we showed that our results are in line with those seen in simulation. Further, to reduce the energy consumption induced by the energy-hungry UWB radio, we investigated two energy saving techniques: deep sleep and preamble sniffing. In particular, we explored the trade-off between current draw and discovery rate introduced by the preamble sniffing, showing that this technique must be configured carefully to identify an acceptable balance for the target application.

Chapter 5

Validation of Indoor Positioning Solution for Automotive

This chapter presents the results of a project with the objective to validate a commercial localization solution for indoor automotive applications. The project was undertaken in collaboration with an external company, which provided the commercial localization system to be evaluated. While not directly related to the primary topic of proximity detection, it uses the same technology exploited in the early sections of this thesis, UWB, as the basis for the validation.

After an introduction in which the scenario and the evaluation system are described, we outline the results, comparing the UWB-based localization to GPS in an outdoor environment and to the commercial system under evaluation in an indoor garage.

The work described in this chapter is in press and will be presented at the 9th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI) in June 2023.

5.1 Introduction

The goal in this project is to define a validation methodology to assess the performance of an absolute positioning system for vehicles, referred to in this chapter as the "commercial V2X system", in GNSS denied environments.

In aid in selecting the proper validation technology and methodology, the external company outlined several requirements, including:

- Less than 1 meter horizontal accuracy in the XY plane (2D localization)
- Sampling rate of 10 Hz, analogous to that of the commercial V2X solution
- Rapid deployment and coverage of critical areas of the test environment
- Avoid interference with 2.4 GHz, for security motivations in the test area

In outdoor scenarios these requirements can be fullfilled with an highly-accurate GPS, e.g., Real-Time Kinematic (RTK), that provides an accuracy of less than 10 cm, depending on conditions [44]. In indoor settings, however, such technologies are unusable due to limitations of satellite coverage, requiring alternative technologies to be considered. An immediate, low-cost solution is to install visual markers on the ground, make a video recording of the vehicle movement and manually identify when the vehicle arrives at each marker. While simple, this technique is open to human error, making it unsuitable as a validation technique. An alternate solution can be formed by using a motion capture system, similar to the one presented in Chapter 2. However, this technology does not satisfy the requirement on the rapid deployment in the testing area, with its lighting requirements and inherit time-consuming installation and calibration. Radio technologies multiple alternatives for localization, e.g., based on BLE [5, 78] or WI-FI [1]. Nevertheless,

these approaches often have accuracy errors greater than 1 m and can generate interference in the 2.4 GHz band, which is strictly forbidden in our project as it represents the working frequency of security system in the testing garage. RFID offers a promising radio technology that may offer accuracy at the level of a few centimeters [96]. Unfortunately, such systems require a large equipment investment. Also significant, some RSSI systems require complex installation, which is not feasible in the target garage environment.

Given our multi-year experience with the UWB radio, and the fact that it meets all the aforementioned requirements, we chose to pursue this technology for the validation system. This choice allowed us to improve our knowledge about this important technology and demonstrate its ranging performance in a different application and scenario.

In this chapter, after a description of the scenario (§5.1.1) and the UWB validation system (§5.2), we evaluate the accuracy of UWB-based localization in an indoor environment (§5.2.3). We then move to an open-air setting where we compare the results of our system to a highly accurate GPS (§5.3.1). Subsequently, we show the final results achieved in the garage, with comparisons between the UWB and the commercial localization systems (§5.3.3).

5.1.1 Considered Scenario

As the target evaluation is for GNSS denied environment, the setting identified for experiment is the ground-level garage at our institution, the map of which is shown in Figure 5.1. The parking lot is composed of two areas, identified by the numbers 1 and 2 in the figure. These two areas are connected by a corridor, while between them a lift area is present. The entrance and the exits are shown with green arrows. Area 1, is wider than Area 2, allowing us to perform several types of maneuvers with the car. It also represents the area with the greater number of parked vehicles during working hours. The lift and stairs areas shown in the picture are highlighted because they represent possible sources of interference for the radio communication technologies used both by the testing system as well as the commercial system to be validated. Specifically, the lift areas are formed by largely metallic elements which are known to induce significant electromagnetic shielding between area 1 and area 2, bringing into question the utility of anchors placed in one area to provide valuable readings for tracking a vehicle in the other area.

In addition, several other aspects make the parking garage a particularly harsh environment for evaluation. Figure 5.2 shows part of area 1, with critical elements highlighted in red. The presence of parked cars, especially during working hours creates reflective waves that can change the electromagnetic propagation properties, making it more difficult to correctly receive the transmitted signal. Moreover, since the number of cars changes during the day, these effects are not constant. Pillars and in general reinforced concrete structures introduce significant signal attenuation and reflections due to the presence of metal inside the structures. The parking lot also has metal pipes in the ceiling that may affect the signal propagation, creating undesired electromagnetic reflections. Overall, the aforementioned elements increase the complexity of the target environment, both for the commercial system as well as for the validation technology. Therefore, as shown later in this chapter, to increase the accuracy of the ground truth system, we performed several experiments to ensure the UWB anchors were deployed specifically to compensate for the challenges posed by environment.

5.2 System Description

This section describes the architecture of the UWB validation system. We illustrate tests performed to evaluate the error of the UWB system in the garage environment

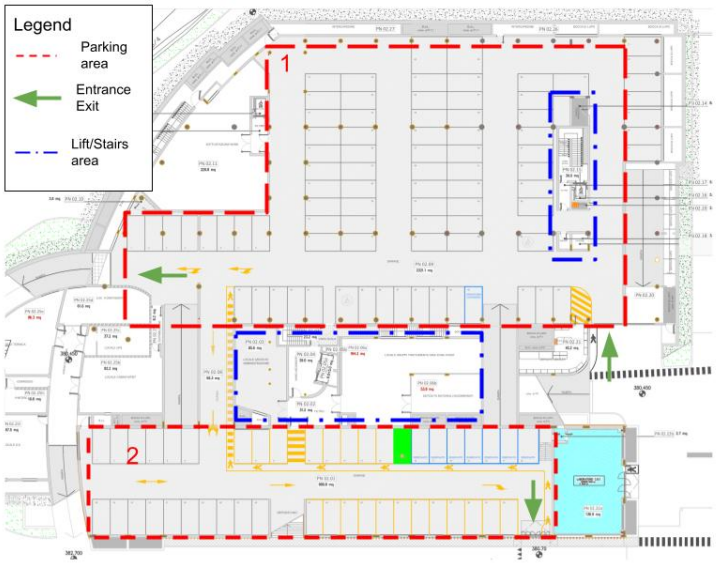


FIGURE 5.1: CAD map of the indoor testing area.

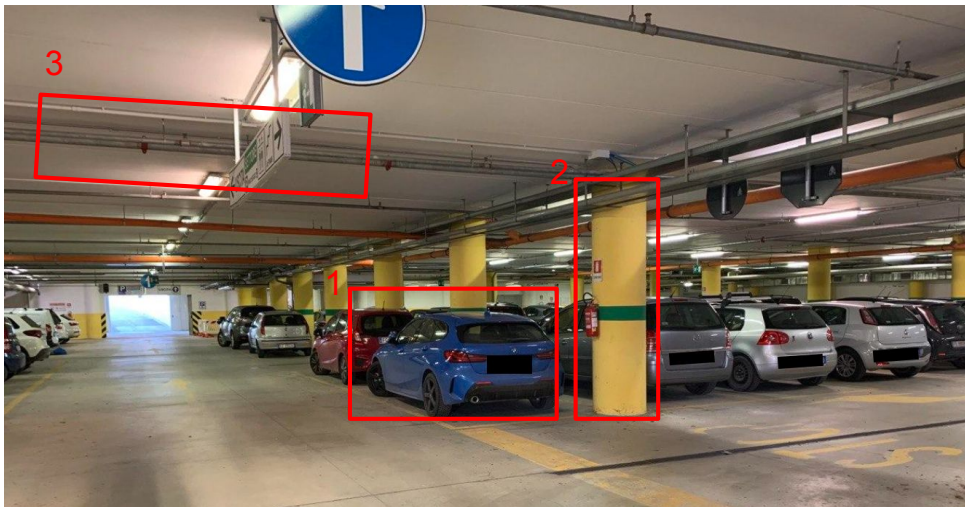


FIGURE 5.2: Photo of test garage highlighting possible sources of interference (1. Parked cars, 2. Concrete pillars, 3. Metal pipes).

and the steps taken to enable comparison between the outputs of the validation system and the commercial V2X system.

5.2.1 System Architecture Overview

The V2X system, provided by an external company, is composed of infrastructure and in-vehicle components. Specifically, multiple anchors must be installed within the indoor environment while an antenna is installed on the roof of the car. An on-board unit inside the vehicle collects data for evaluation. Since the system works with World Geodetic System 1984 (WGS84) coordinates, each anchor is associated to GNSS coordinates in latitude and longitude. As this system was not

under our control, we do not provide further information[1], except to note that the system is advertised specifically for indoor and outdoor environments. As previously stated, our project focused on the indoor evaluation.

The UWB-based localization system was composed by:

- Fixed anchors (minimum: 3. maximum: 30)
- Active tag, installed on top of the car
- Tablet with the QORVO DWM1001 Two-Way-Ranging Real Time Location System (DRTLS) management application
- Computer, to collect logs.

This system is based on hardware and software provided by QORVO [84], for use with the DecaWave DWM1001 platform. This platform contains both the UWB and BLE radio. The firmware considered for this application is the DWM1001 factory firmware version 2, released by the same company. Using this firmware, the maximum data rate is 10 samples per second, where the samples are XYZ locations.

Figure 5.3 shows the overall architecture of the UWB system: the DWM1001 devices must be programmed using the QORVO DRTLS Manager application, in particular defining the role (active tag, anchor, passive tag) and setting the positions in XYZ coordinates for the anchors. The DRTLS application exploits the Bluetooth connection between the tablet/smartphone and the devices to modify the parameters of each single device. In particular, the anchors are fixed devices that allow the active tag to compute its position using a location engine based on triangulation. More details on the protocol can be found in Section §5.2.2. Once the anchors and the tag are programmed, they can be deployed on the field and the XYZ positions measured at the active tag are sent via Universal Asynchronous Receiver-Transmitter (UART) to a PC, where the logs are saved.

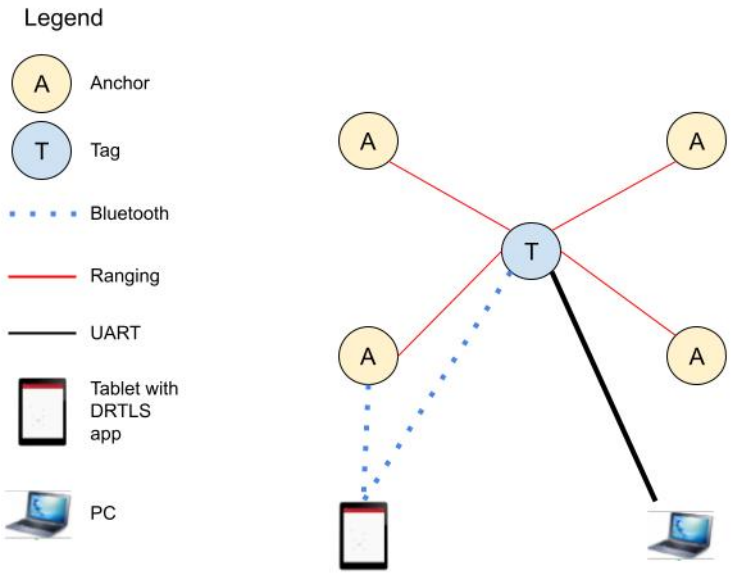


FIGURE 5.3: Architecture overview of the UWB test system.

5.2.2 UWB Evaluation System Setup

In the following subsections, the setup of the UWB system is detailed, from the anchor installation to the description of the firmware used.

[1]The name of the company involved and the information about the commercial V2X system are anonymized for explicit request from the company itself.

UWB Anchor Installation

To guide the anchor installation, we relied on a computer-aided design (CAD) map of the garage area. An evaluation using a laser range finder of the CAD map revealed an accuracy of approximately 5 cm in the location of structural elements such as pillars and walls. Therefore, we chose to locate UWB anchors at identifiable points on the CAD map. e.g., on the support columns, deriving their XY location directly from the CAD. To reduce interference caused by the presence of vehicles, anchors were positioned as close as possible to the height of the antenna on the test vehicle, approximately 2 meters.

Figure 5.4 shows an example of an UWB anchor, deployed on a pillar using an elastic band from a waist bag that offers a stable support without damaging the pillars. When anchors were placed "freely" in the environment or when it was impossible to exploit elements in the environment such as pillars, we use a microphone stand with an UWB anchor mounted at a height of approximately 2 meters, as shown in Figure 5.5.

FIGURE 5.4: Mounting mechanism for UWB anchors on pillars in the garage environment. Different from the figure, the height was approximately 2 m.

Time Synchronization

The computer collecting data from the UWB evaluation system must be time-synchronized to the commercial V2X system to allow the logged location data to be correlated. This is accomplished via NTP, using the commercial V2X on-board unit as the NTP server and running the client on the computer supporting the UWB data collection. The on-board unit of the commercial V2X synchronizes its internal clock using the GNSS signals when they are available. The client application, running on a dedicated PC, is configured to synchronize its system time to that of the on-board unit using the Ubuntu systemd-timesyncd service via Ethernet. This PC also collects the UWB logs by means of the UART connection. The data items collected on the PC are time stamped on arrival, with the timestamp coherent with the common time defined by the on-board unit.

Antenna Offset

For practical reasons, the antenna of the commercial V2X system and the UWB device (used for measuring the UWB locations) cannot be at the same, exact position

FIGURE 5.5: UWB anchor on a movable microphone stand at approximately 2 m height.

on the roof of the car. The GNSS/V2X antenna is located along the central, lengthwise axis of the test vehicle and we choose to mount the UWB device along the same axis.

Figure 5.6 shows the UWB device on the car, attached using a box mounted on a magnetic anchor. The distance between the two antennas is measured manually. This offset is used to adjust the UWB traces. This is done calculating an average direction of the test vehicle over a variable size set of locations. For each measured location, the measured offset is applied along the model. The number of points used to calculate the model depends on the speed of the vehicle. We experimentally determined that a set of size 5 points is reasonable for the speeds expected during experiments. It should be noted that this procedure correctly applies the antenna offset only when the car is in motion.

Coordinate Conversion

Once the V2X data and the UWB data are collected, several transformations are required to allow them to be directly compared. With reference to Figure 5.7: The V2X system logs are converted from Latitude and Longitude coordinates to Cartesian coordinates in the CAD reference system; The UWB logs, referred to the UWB reference system defined by the DRTLS manager app, are first modified to account for the antenna offset described in the previous section. Subsequently, we convert the location from the UWB reference system to the CAD reference system.

Once both the logs are in the same coordinate system, they can be compared. The conversion from the V2X system (or the UWB system) to the CAD reference system is done using a triangulation script. In particular, three reference points identified with WGS84 (latitude and longitude), UWB and CAD coordinates have been selected during the initial inspection of the garage.

Finally, the data are compared using the analysis as described in §5.3

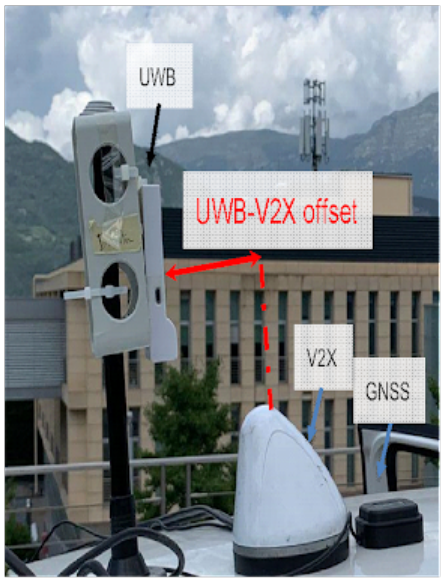


FIGURE 5.6: UWB antenna and V2X antenna mounted on a test vehicle. The antenna offset is shown. Values of 5 to 9 cm were typical during experiments.

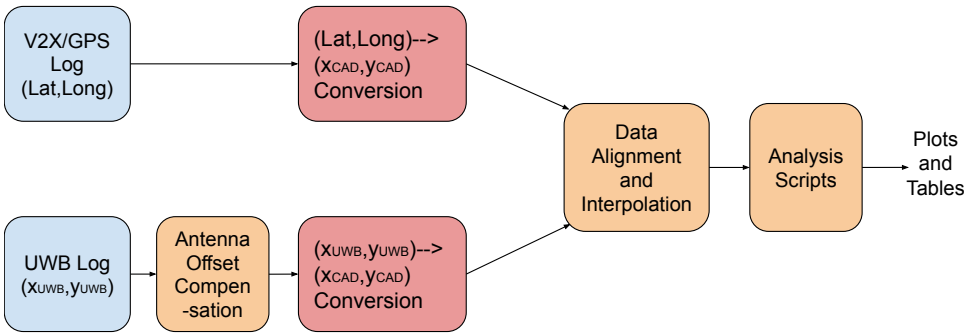


FIGURE 5.7: Data processing pipeline enabling comparison between V2X latitude/longitude logs and the UWB XY logs.

Protocol Details

For completeness, we offer a brief description of the DRTLS. In particular, we focus on the localization component, ignoring other aspects of the protocol, e.g., channel access, collision avoidance, etc. All the information included here are taken from [25] .

The nodes (tags and anchors) operate using a repeating "super frame" structure of 100 ms duration. This structure is shown in Figure 5.8. The super frame starts with 30 Beacon Message (BCN) slots, in which each anchor transmits a packet containing its XYZ position (positions are constant in our scenario), followed by two Service slots (SVC). There are additionally 15 TWR slots used for the (mobile) tags to initiate TWR exchanges with the anchors (details below). Each TWR slot can be associated with a specific tag, which uses the slot to perform ranging with 3 or 4 anchors. Both the anchors and the tag are synchronized; therefore the tag will listen during the BCN phase to receive information from all the anchors in range. For the remainder of the super frame the tag is in idle mode, while the anchors perform other operations such as synchronization.

During each super frame period:

- Each anchor is associated with a unique BCN ID, and during each BCN the

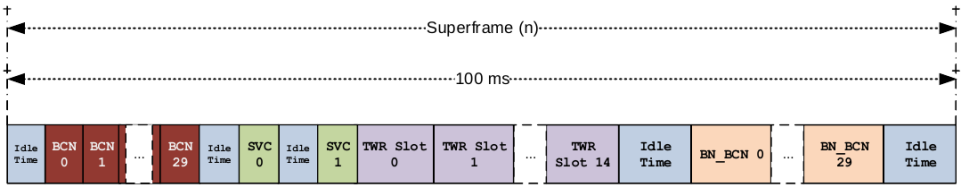


FIGURE 5.8: Super frame of the DRLTS system [25].

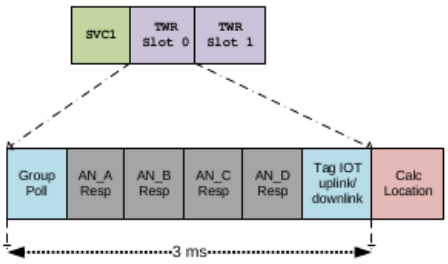


FIGURE 5.9: Details of a single two-way ranging slot in the DRLTS application [25].

associated anchor transmits its position in the UWB coordinate system. The position is defined during the deployment phase using the DecaWave DRTLS Manager application as mentioned above;

- The tag listens for all the BCN from all the anchors in range and saves the received information;
- During the TWR phase (shown in Figure 5.9), the tag chooses 4 anchors based on the positions received during the BCN phase. The algorithm chooses three or four anchors that are both close and that define a polygon around the tag with one anchor per quadrant. The tag is initialized with 0,0,0 as the default position. With reference to Figure 5.9, the TWR exchange is as follows:
  - The tag sends a group POLL message, with the 4 IDs of the selected anchors and the order in which the anchors must reply.
  - Each anchor will reply with the RESPONSE message;
  - Once the tag receives at least 3 non-corrupted ranging messages, it calculates the distance from each anchor;
  - The tag uses the 3 or 4 calculated distance values to compute the XYZ position using its proprietary localization algorithm.
  - Note that a vehicle travelling at 20km/h will travel approximately 1.7 cm during the 3 ms required to collect the ranging information from the anchors.
- The tag sends the computed XY position to the tablet/smartphone and/or to PC via UART where the locations are stored.

5.2.3 UWB Accuracy Evaluation in the Garage

To evaluate the accuracy of the UWB system in the garage environment, we performed six experiments with different size areas (denoted Large, Medium and Small). All experiments were conducted during the daytime and therefore with multiple parked cars in the garage. In half of the cases, we designed the test area such that it did not include any parked cars and there was direct line of sight

between the anchors and the test device, with the objective to reduce the multipath effects caused by the parked vehicles. The result, however, is a more narrow rectangular test area, as seen in the leftmost image of Figure 5.10. In this setup, all anchors, installed at the corners of the test areas, were mounted on the microphone stand. The second set of experiments, shown on the right of Figure 5.10, enlarged the width of the rectangle by placing two of the anchors on pillars behind a row of parked vehicles. The other two anchors were placed using the microphone stands.

For each test, we identified 3 locations inside the areas (Corner, Center, Side), and precisely measured these locations with a laser range finder. These locations were modified for each test area to evaluate the three general zones (corner, center and side) of the rectangular test area. Tests were performed using a portable UWB node mounted on the microphone stand, placed at each location. At each location, XY location measurements were collected using the UWB localization tool, collecting data for three minutes at each location. An example area with the three points is depicted in Figure 5.11.

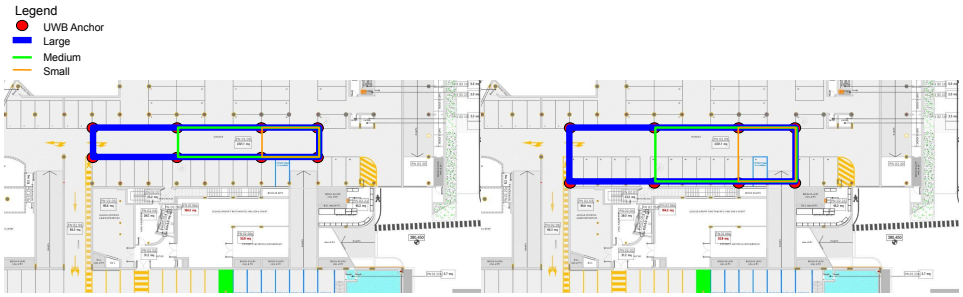


FIGURE 5.10: Left: narrow test, small (orange), medium (green) and large (blue) areas. Right: wide test, small, medium and large area. In all cases, anchors are placed at the corners with the antenna facing the vertical axis.

FIGURE 5.11: Wide test, small area, showing the three sample points (corner, center and side).

Evaluation

We separately evaluate the root mean square (RMS) error on the X and Y axes as well as the RMS of a combined measurement of the distance-error, obtained by calculating the straight line distance between the ground truth location and the measured (X,Y) location.

In most cases, the error is within 30 cm. We note that the x-axis error is in general larger than the y-axis error as the x-dimension of the rectangle is larger than the y. Interestingly, the difference in error across the different size test areas is not significantly different. Further, the inclusion or not of cars in the test area does not have a significant effect, most likely because the anchors and test antenna are above the level of the vehicles, reducing the vehicle occlusion.

TABLE 5.1: Wide test area, with cars inside rectangle defined by the anchors. Root Mean Square error in meters reported as the difference between ground truth and measured values across all experiments.

	Small Area			Medium Area			Large Area		
	Corner	Center	Side	Corner	Center	Side	Corner	Center	Side
RMS - X axis	0.21	0.29	0.26	0.27	0.12	0.63	0.24	0.09	0.48
RMS - Y axis	0.02	0.11	0.09	0.10	0.22	0.24	0.01	0.15	0.21
RMS - distance	0.22	0.31	0.28	0.29	0.25	0.67	0.24	0.18	0.53

TABLE 5.2: Narrow test area, with no cars inside rectangle defined by the anchors. Root Mean Square error in meters reported as the difference between ground truth and measured values across all experiments.

	Small Area			Medium Area			Large Area		
	Corner	Center	Side	Corner	Center	Side	Corner	Center	Side
RMS - X axis	0.20	0.24	0.15	0.30	0.29	0.22	0.24	0.40	0.28
RMS - Y axis	0.04	0.12	0.06	0.03	0.07	0.11	0.12	0.20	0.10
RMS - distance	0.20	0.27	0.16	0.30	0.30	0.24	0.27	0.45	0.30

These tests, even if quite simple, demonstrated that the UWB system provides the necessary accuracy for the requirements of this project, despite the harsh indoor environment.

5.2.4 Final UWB Anchor Locations in Garage

Based on the results above and some additional tests performed with larger areas of varying shapes, we identified two sets of anchor locations to use for the final experimentation. These locations and the areas they cover are shown in Figures 5.12 and 5.13. Note that the orange arrows indicate the orientation of the anchors (more specifically the antenna orientation) and the blue arrow indicates the direction of the car's movement. The green lines instead represent the path that the car covers, depending on the experiment. Figure 5.12 represents the case in which the car goes straight through the main path of the garage. Figure 5.13 shows a more complex path, with 18 anchors installed.

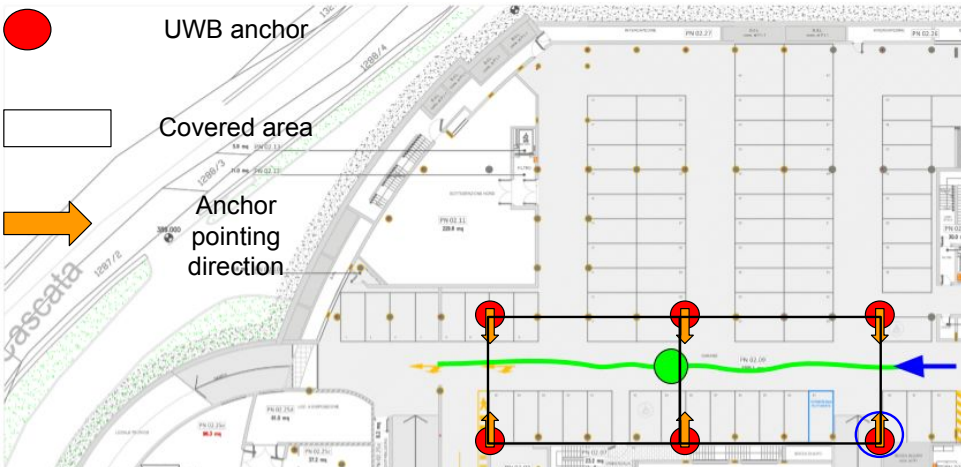


FIGURE 5.12: Test area 1 showing 6 anchor locations. Orange arrows indicate antenna orientation.

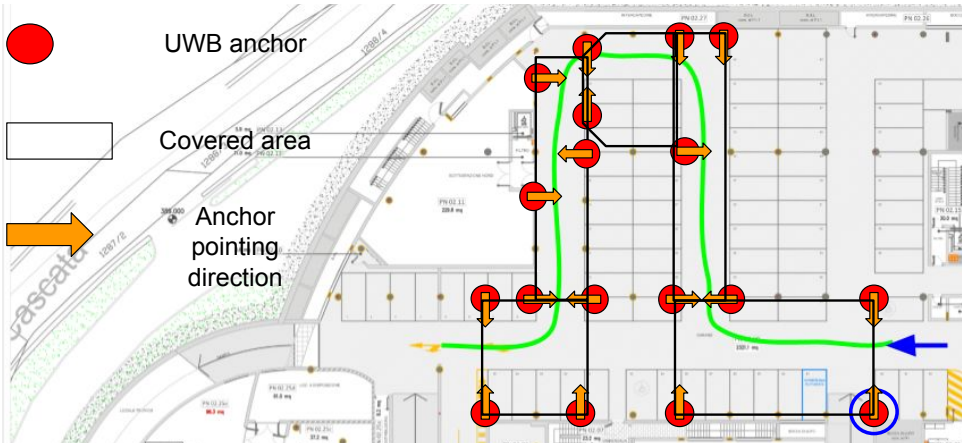


FIGURE 5.13: Test area 2 showing 18 anchor locations. Orange arrows indicate antenna orientation.

5.3 Experiments and Results

This section outlines the set of experiments performed to evaluate the commercial V2X system against the UWB validation system. In addition to the final indoor traces, we detail additional experiments performed to demonstrate the quality of UWB system and provide evidence for why this technology can be used as a valid comparison against the commercial system, we divided the experiments into three different trials, each with a different goal:

- Section §5.3.1 shows experiments conducted in an outdoor environment in which UWB and a GPS system have been used to record locations. The scope was to compare the results of the UWB system with the GPS, considering the second as the ground truth. This offers quantitative accuracy measurements for the UWB localization system.
- Section §5.3.2 shows experiments involving only UWB. The scope was to understand whether the UWB traces were coherent with respect to the CAD map.
- Section §5.3.3 details the final experiments in which the UWB and the V2X technologies are compared. The aim was to assess the quality of the commercial V2X system, taking into account the accuracy of UWB demonstrated by the previous experiments.

5.3.1 Outdoor Experiments

UWB System Setting

The experiments were conducted on the terrace in front of the north building at our institution. Figure 5.14 shows the environment, together with markers that identify the position of UWB anchors.

The UWB anchors were attached to the lampposts labeled from 1 to 8, except for number 5, mounted on a microphone stand. Each position was identified by a WGS84, UWB and CAD coordinates, making it possible to calculate the CAD coordinates of the UWB tag with trilateration as explained in Section §5.2.2, using as reference points 2, 3 and 6. In each position, the UWB anchor was attached at approximately 2 m height as depicted in Figure 5.15a and 5.15b.

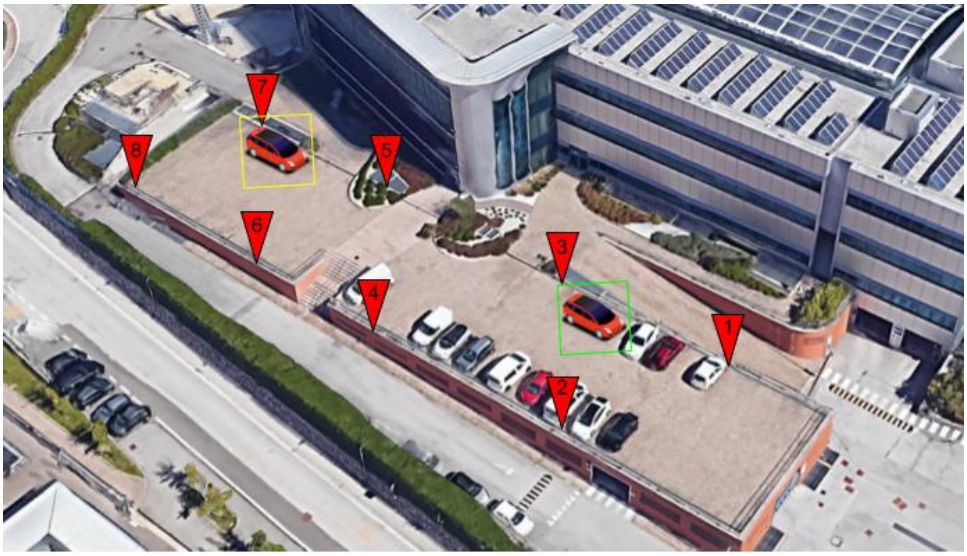


FIGURE 5.14: Google Earth view of the experiment site. The red arrows indicate the UWB anchors positions, while the two red cars with the colored squares around, the starting positions of the two experiments. (during the experiments fewer cars were parked on the terrace).

(A) Example of an UWB anchor attached to a lamppost.

(B) Close up image of UWB anchor attached to a lamppost.

FIGURE 5.15: Example of UWB anchor installed on a lamppost.

Experiment Description and Results

The goal of these experiments was to demonstrate the quality of the UWB technology by comparing it to a GPS system that provides a WGS84 position at 10Hz with an error of approximately 10 cm. Each *experiment* was conducted 5 times (*trials*). For each experiment, 8 UWB anchors were installed. On the roof of the car the UWB tag was positioned on the central axis of the car, 6 cm in front of the GPS antenna. The following describes the experiments and results.

Timestamp Offset Evaluation

Before the description of the experiments, we illustrate a problem common across all experiments.

Upon evaluation of preliminary data collected in the outdoor areas, we noticed that, despite the time-synchronization between the commercial V2X system and logging PC, there was still a time offset between locations.

In Figure 5.16 two examples of the same trace are shown, to depict the effect of the *index offset*. In both figures, the data are downsampled, i.e., there are few samples than the originals tracks in order to have a clearer visualization. The black circles are GPS data, while red circles are UWB data. In Figure 5.16a each GPS sample is associated to the closest (in time) UWB sample. This association at first instance is reasonable since the GPS and the UWB systems are time-synchronized. In this case, we consider to have a *index offset* equal to zero, i.e., to each GPS sample we associate the closet (in time) UWB sample. However, between the two samples there is a non-negligible distance (green line), suggesting a possible disalignment in time between the systems. Therefore, in Figure 5.16b, we substituted the closest UWB samples in time, with the samples that minimize the mean arithmetic distance, w.r.t. the GPS data. Doing so, each GPS sample is then associated to a closer (in distance) UWB sample, showing that a difference in time between the UWB and GPS system exists, even if are time-synchronized. To do so, in post processing we tried different index offsets, to find the one that minimized the distance difference. We reference to an index offset, since the samples are discrete values sampled each 100 ms by both the systems: for instance an index offset of 2 indicates that the point at index=1 of the GPS trace is compared to the UWB trace at index=3. Therefore, in calculating the mean arithmetic distance between the two systems, each GPS sample is associated to the UWB sample around 200 ms later.

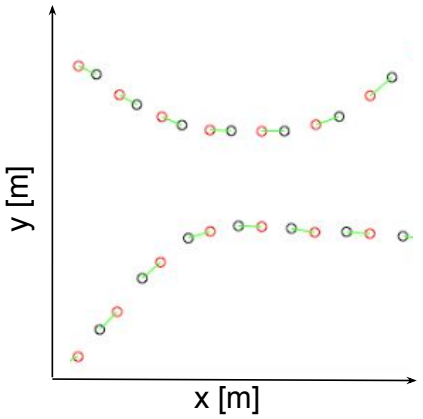
Practically to determine the time offset between the recorded WGS84 locations and the recorded UWB locations, for each trial, we time-aligned the traces according to the recorded timestamps, then calculated the distance error between the index-aligned points. We then calculated the same distance error with index offset values between -10 and 10, corresponding to time-offsets of -1 second to +1 second (-1000 ms to 1000 ms). Across all traces, the index with the minimum error is 3, corresponding to 300 ms. The calculated error for multiple offsets for a single trace is reported in the tables for each experiment.

Eight Shape

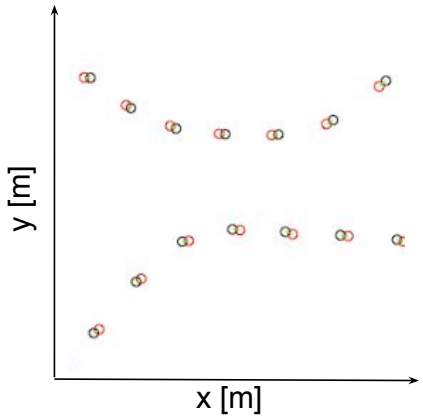
Description: In our first experiment, the entire length of the terrace is covered by the path of the car, taking care to stay within the UWB coverage area, i.e., without driving past the borders defined by anchor 1 and anchor 8. The car started between anchor 3 and 1, in the position and direction shown with the red car and the green square in Figure 5.14. The car traveled the terrace and closed the loop near the starting position. The speed of the car was approximately constant at 10 km/h: a reasonable parking lot speed.

Target: evaluating the accuracy of the UWB in a relatively large area, considering also the handover between the anchors while the tag moves inside the area.

Results: One sample trace is shown in Figure 5.17. For each trial, we apply in pre-processing the antenna offset to the collected UWB data, effectively shifting all UWB points 6 cm “backward” with respect to the direction of the vehicle.



(A) GPS (black) and UWB (red) samples from a 8-shape track, without the index offset compensation.



(B) GPS (black) and UWB (red) samples from a 8-shape track, with the index offset compensation.

FIGURE 5.16: Same part of the same experiment, with and without the index offset compensation. For visualization purposes, the tracks are downsampled.

TABLE 5.3: 8-Shape experiment result summary.

	Sampling rate 10Hz		Artificially increased sampling rate	
Trial Index (No.)	Index Offset (No. of samples)	Mean Distance (m)	Index offset (No. of samples) minimizing mean distance	Mean Distance (m)
1	-3	0.113	-35	0.104
2	-3	0.102	-34	0.097
3	-3	0.097	-34	0.092
4	-3	0.099	-34	0.093
5	-3	0.109	-34	0.105

Table 5.3 reports the index offset between the UWB and GPS traces that minimizes the average distance between the GPS and UWB points over the entire trace. It also shows the mean difference for the entire trace when this offset is applied. This index offset primarily accounts for the delays introduced during the UWB sample acquisition pipeline. Nevertheless, the sampling time of the UWB and GPS traces are not synchronized in time. With a car traveling at a speed of 10 km/h and sampling every 100 ms, the distance between two samples in a single system is 28 cm. To compensate for this, we artificially increase the sampling rate of the GPS trace, adding 10 equi-distant points along a line connecting each pair of points in the original trace. We then identify the index offset for this artificially upsampled GPS dataset with respect to the UWB dataset and report this new index and the corresponding mean distance in the table.

Table 5.3 shows that for the outdoor experiments, the difference between the GPS and the UWB locations is very small, on the order of 10 cm. We note that the mean distance reported with the upsampled GPS data is smaller than the 10 Hz data.

It is important to note that while most of the time the two traces, GPS and UWB, are very close to one another, at one point the UWB data deviates significantly from the GPS. As seen in Figure 5.18, the UWB trace changes trajectory, increasing the distance between the two traces in that area. In this area, the car was driving very

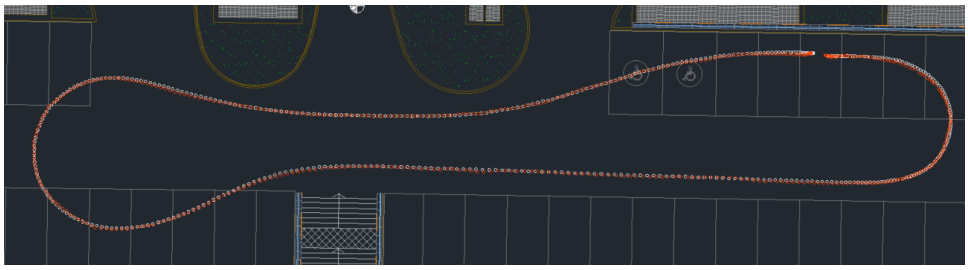


FIGURE 5.17: A full trial of the 8-shape experiment. White and red dots are RTK and UWB data, respectively.

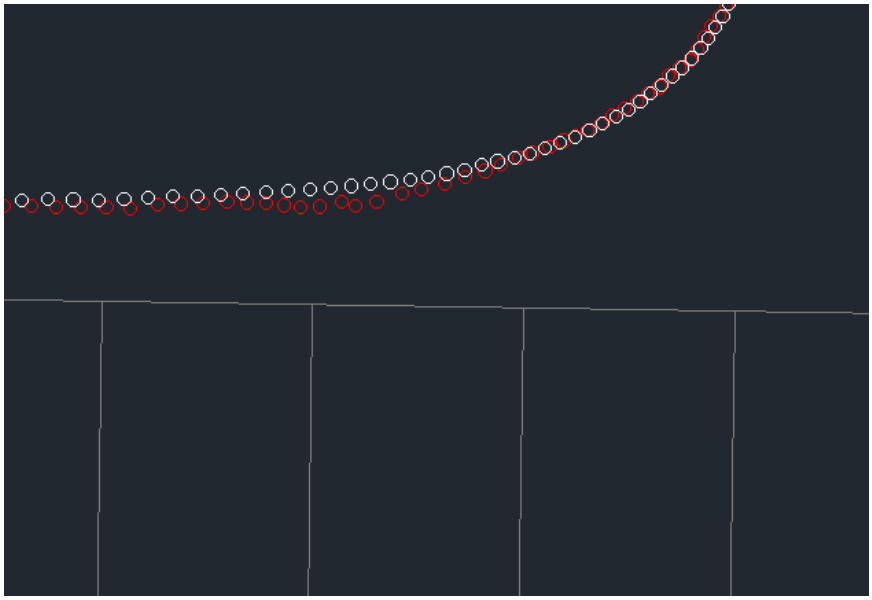


FIGURE 5.18: Detail with UWB deviation in one 8-shape trial. White and red dots are RTK and UWB data, respectively.

close to the parked cars, which likely creates interference in the UWB measurements. This is further exacerbated by the fact that one of the UWB anchors was behind those cars (marked with 2 in Figure 5.14).

We conclude this section with a plot that shows the measured difference over time between the GPS and the UWB values. The plot of Figure 5.19 shows a single trace with most differences well below 0.2 m. Analysis of the data reveals that the spikes seen here around indices 100 and 200 occur when the car is turning the corner at the edges of the two traces. This area is nearest to the edges of the UWB coverage area and likely has larger error. In other traces, e.g., Trace 05, we see an even larger spike just before index 200. In this case, the problem arises because the original UWB dataset has a gap in the data of multiple (3) samples. Although we applied a mechanism to interpolate between these gaps, this mechanism is by definition an approximation and thus generates the larger differences seen in the plots. Despite these larger errors, the difference is quite reasonable throughout the experiments.

Small Circle

Description: In our second experiment, a restricted area of the northern portion of the terrace is traversed by the car, with the start at the stop between 7 and 5, as depicted with the red car and the yellow square in Figure 5.14. The car traveled from the start to the stop moving in a circle, at a speed less than 10 km/h, due to the relatively small area available for the manoeuvre. There were no cars parked in that area of the terrace. As the area was small, the UWB positioning system always

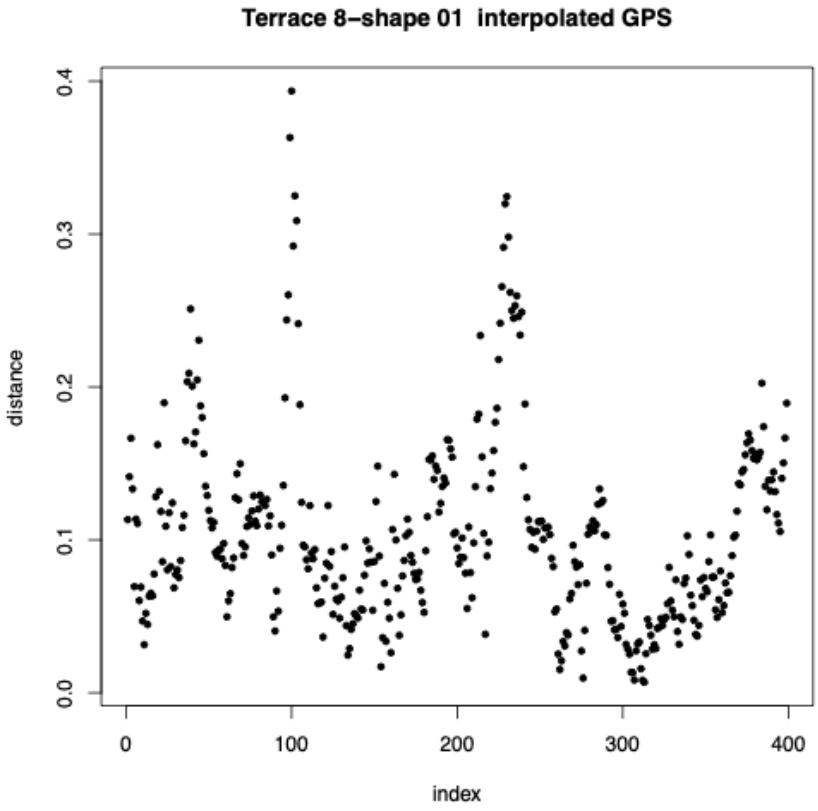


FIGURE 5.19: Distance difference in meters between the GPS and UWB traces for the duration of trace 1. The index corresponds to the 10Hz samples of the UWB trace.

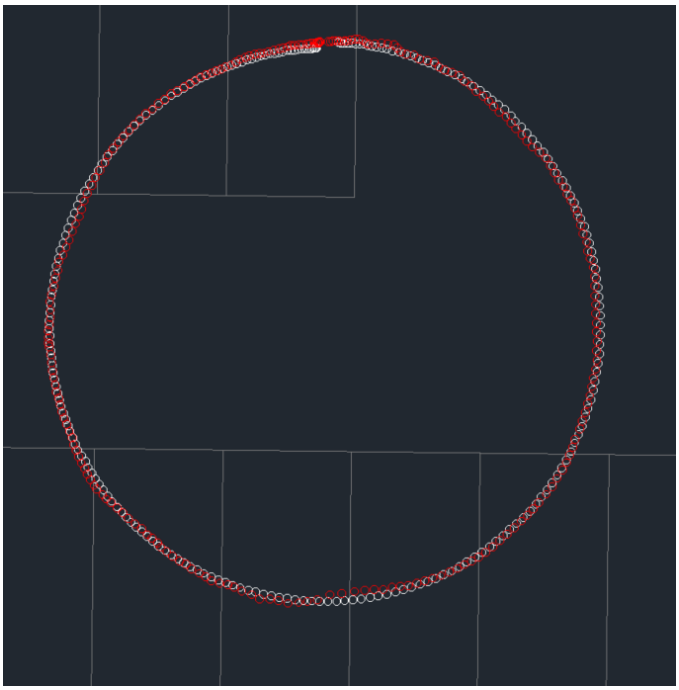


FIGURE 5.20: A full trial of the small circle experiment. White and red dots are RTK and UWB data, respectively.

used the same four anchors for the location computation: 5,6,7,8 (Figure 5.14).

TABLE 5.4: Circle experiment result summary.

	Sampling rate 10Hz		Artificially increased sampling rate	
Trial Index (No.)	Index Offset (No. of samples)	Mean Distance (m)	Index offset (No. of samples) minimizing mean distance	Mean Distance (m)
1	-3	0.083	-32	0.082
2	-3	0.084	-31	0.080
3	-3	0.087	-31	0.085
4	-3	0.088	-32	0.086
5	-3	0.084	-31	0.079

Target: Testing the UWB system in a smaller without any other vehicles.

Results: The plots in Figure 5.20 depict the traces with both the GPS and the UWB data with the 6 cm offset applied, in white and red respectively. As in the previous experiments, the results demonstrate that UWB achieves good localisation performance, with a distance in mean under 20 cm with respect to the GPS data. Table 5.4 shows both the time offset and the mean distance difference achieved in this case: the time offset is around 3 samples as in the previous experiments. Overall, the performance is better when compared to the previous experiment, probably because of the smaller testing area and the reduced speed.

As in the previous 8-shape experiment, traces in this experiment also show deviations between the RTK and UWB. In Figure 5.21 the deviation GPS becomes significant only during the curve when around 10 UWB samples are more distant from the GPS trace. While in the previous case, the problem was likely due to the presence of the parked cars near one of the UWB anchors, this issue likely arises as the car was near the border of the coverage area. As the antennas were pointed

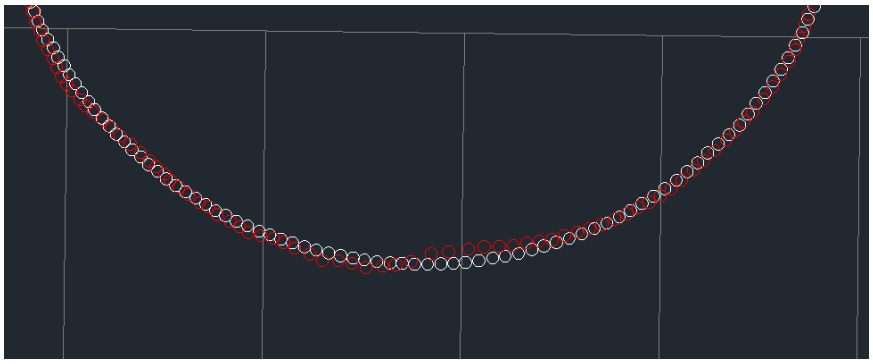


FIGURE 5.21: Detail with UWB deviation in one small circle trial. White and red dots are RTK and UWB data, respectively.

toward the center of the terrace, the errors in the measured distances in these areas have larger errors. This is also visible in Figure 5.22 where the difference over time is shown. Around index 130, the car is near the north side of the terrace while at the beginning and end the car was near the top of the terrace (nearest to the building). In any case, we note that the error is below 0.2 m in all cases.

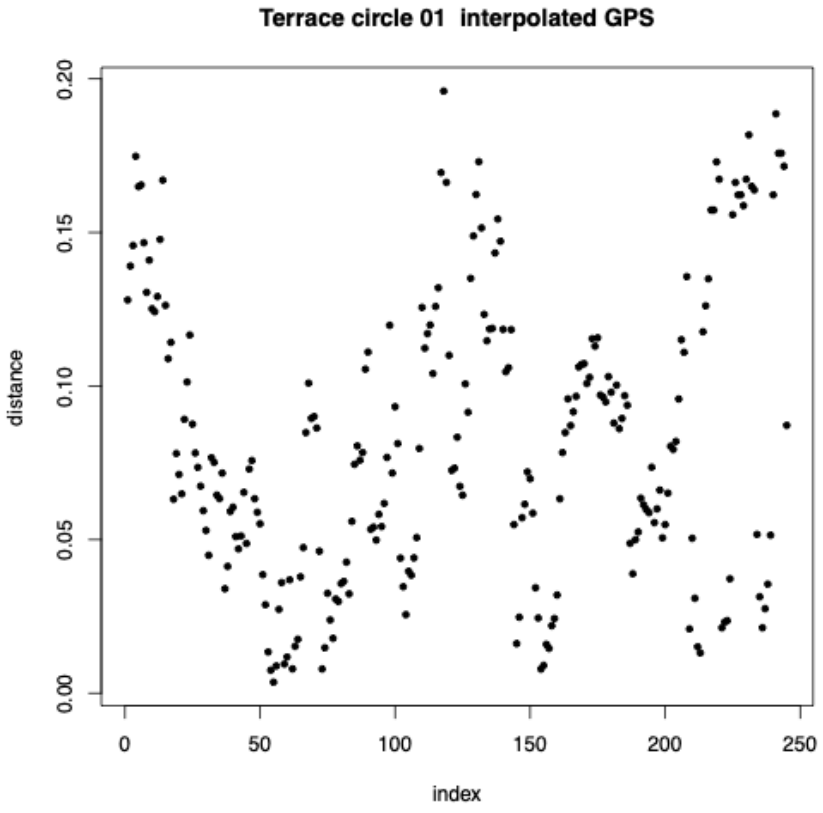


FIGURE 5.22: Distance difference in meters between GPS and UWB for the first circle experiment.

5.3.2 Preliminary Indoor Experiments

Before experimenting with the vehicle in the garage, we performed a set of experiments with individuals walking in the covered area.

Experiment Description and Results

The preliminary experiments in the garage were completed by an individual carrying a device at nearly 2 m height and walking in the garage. Only UWB data were collected as the scope of the experiment was to confirm the coherence of the UWB coordinates and the position conversion scripts with respect to the CAD map. For this reason we walked close to the lines delimiting the parking spaces to evaluate whether the collected traces were coherent with the CAD map. It is important to mention that these experiments were carried out before the definition of the final configuration of the UWB anchors, and some anchor positions were modified after evaluating the traces collected in these experiments. Information regarding these changes is detailed in the experiment results section below.

Finally, in these walking experiments we use a BLE connection, with a *listener* device between the UWB tag and the UART connection to the PC. This configuration was changed for all the other experiments, with the UWB tag directly connected to the PC as explained before.

Full Walk

Description: the full coverage anchor configuration was used and the user walked from the start to the stop points moving along the borders of the parking spaces, without stopping. This experiment used only 16 anchors and one anchor was at a 90 degree angle w.r.t. Figure 5.13.

Target: Verify that the border of the parking spaces were respected and confirm the functioning of the UWB system and the CAD position conversion scripts.

Results: Figure 5.23 shows a sample trace. The recording started on the right side of the figure and continuously recorded until the stop position at the left of the picture. From the figure it can be seen that, even if the tracks are slightly noisy due to the UWB technology and the irregular movement of the person carrying the node, the recorded track respects the borders of the parking spaces, showing that the conversion script used to transform UWB coordinates into CAD coordinate is working as expected. It is also important to note the noisier portions of the path, highlighted with the red circles. To compensate for the problems manifested, we modified the configuration to that shown in Figure 5.24, adding two anchors and changing the direction of a third. Additional experiments performed in the trouble spots confirmed that our changes improved the results.

Snake Walk

Description: Using the anchor configuration shown in Figure 5.12, the user with the tag walked from the start to the end without stopping and traveled the main path of the garage, trying to walk as close as possible to the borders of the parking spaces.

Target: As in the previous experiment, the target was to demonstrate the correct functioning of the CAD position conversion scripts.

Results: Figure 5.25 shows the results. In this case, the resulting track is smooth, with no noisy areas, showing that the position conversion scripts work correctly and the straight path anchor configuration was reasonable. Therefore, no anchors were added, moved, or rotated.

5.3.3 Final Indoor Experiments

We next move on to the final experiments, comparing the commercial V2X system to the UWB positioning system.

FIGURE 5.23: Trace recorded during the full walk experiment.

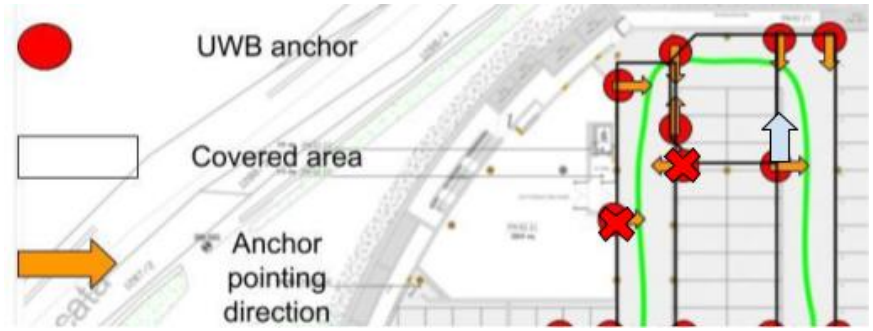


FIGURE 5.24: Old configuration, zoomed view. Two fewer anchors (red crosses) and one anchor pointing in a different direction (blue arrow).

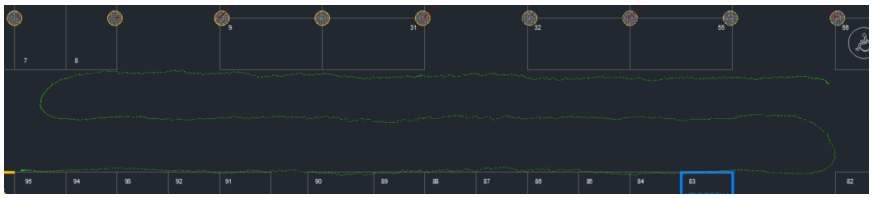


FIGURE 5.25: Track recorded during the snake walk experiment.

UWB System Setting

The UWB system settings are those depicted in Figures 5.12 and 5.13 depending on the type of experiment. The UWB tag used for positioning was installed on the roof of the car as described before with an offset of 6 cm.

Experiment Description and Results

The target of the final experiment was to compare the UWB and the V2X system. At the time of the experiments, the parking lot was full. Before each experiment, the test vehicle was driven outside the garage for sufficient time that the GPS signal was acquired. Upon entering the garage, the car turned left, then paused for approximately from 5 to 10 seconds to allow the UWB system to initialize.

Simple Path

Description: The car started at the right side of Figure 5.26 and stopped near the point at the the middle of the final two UWB anchors, positioned at the left side of the image. The car traveled at a mean speed of approximately 10 km/h. The UWB anchor configuration was that shown in Figure 5.12.

TABLE 5.5: Index offset and error for the indoor Simple Path without Stop.

Trial Index (No.)	Sampling rate 10Hz		Artificially increased sampling rate			
	Index Offset (No. of samples)	Mean Distance (m)	Index offset (No. of samples) minimizing mean distance	Mean Distance (m)	Index offset -3	Mean Distance (m)
1	14	0.805	158	0.799	-31	3.834
2	-5	0.674	-57	0.662	-31	1.167
3	2	2.635	20	2.629	-31	3.072
4	-1	1.802	-12	1.800	-31	1.921
5	3	3.264	31	3.262	-31	3.869

Target: Comparison of the two systems in a simple scenario.

Results: Figure 5.26 shows an example of the UWB and V2X traces in red and green respectively. Despite the simplicity of the experiment, we see clearly that the traces deviate, in particular at the beginning where the starting points are more than 2 meters apart. The difference reduces during the experiment as the V2X trace moves closer to the UWB. This behavior suggests the presence of a systematic offset in the commercial V2X system, however, by evaluating multiple V2X traces, this hypothesis is not confirmed.

Figure 5.27 shows all five V2X traces for the Simple Path without Stop. Here we see that the traces are widely different from one another, with approximately four meters difference between them in the worst cases. On the other hand, the UWB traces shown for the same experiments at the bottom of Figure 5.28 are coherent, showing the car moving consistently down the middle of the aisle. The small differences between the UWB traces are well within the tolerances expected with a human driver, especially considering that the start and stop positions changed slightly on each iteration.

To complete the evaluation of these traces, we performed the same evaluation as the preliminary terrace experiments, comparing the positions reported by the

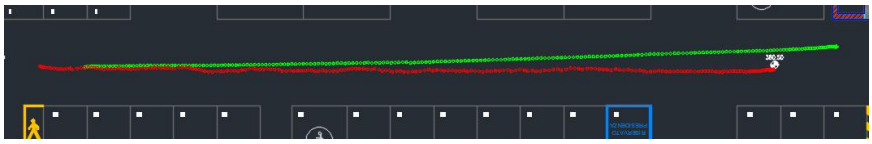


FIGURE 5.26: Single trace of the simple path with V2X in green and UWB in red.

FIGURE 5.27: In green, all the V2X traces.

UWB system to those reported by the commercial V2X system. As before, we apply the antenna offset to the UWB traces and perform the evaluation using the coordinates converted to the CAD coordinate system. To compensate for the time difference between the measurement of the data points, we calculate and apply the index offset that minimizes the difference between the traces for both the original UWB and V2X traces. We then artificially up-sample the V2X trace, adding 10 points between each pair of points, then calculate the new index (time) offset.

As expected based on the visual observation of the traces, the differences between the corresponding traces is much larger than we observed on the terrace between the GPS and UWB, and in one case, above 3 m. In Table 5.5 we see that while the calculated time index of the terrace experiments was consistent at -3, the calculated index offsets that minimize the difference vary widely on this dataset. While this offset minimizes the distance between the points, placing the commercial V2X system in the best light, it is unlikely that the traces were as different as the index values indicate, given that we had time synchronized the devices collecting the traces, just as in the terrace experiments. Therefore, we also show the mean distance between the traces, assuming an index offset of -31, as we saw on the terrace. Even in this case, however the differences remain quite large, as depicted in Figure 5.29.

Simple Path with Curve

Description: The path is represented by the green line in Figure 5.13. The full coverage UWB anchor configuration was used.

Target: Compare the two systems in a realistic parking lot scenario, in which a car enters the lot, searches for a parking spot, then drives to the exit.

Results: Figure 5.30 shows the commercial V2X and UWB traces of one trial of this experiment. As previously shown, the V2X trace has some points that are unreasonable, e.g., at one point the trace overlaps with the parking spaces (which were filled with vehicles at the time of the experiment). Moreover, in these traces the V2X system was inconsistent across experiments, while the UWB system was, again, quite consistent despite the challenges of the garage environment. The traces of all iterations for a single technology are shown in Figures 5.31 and 5.32.

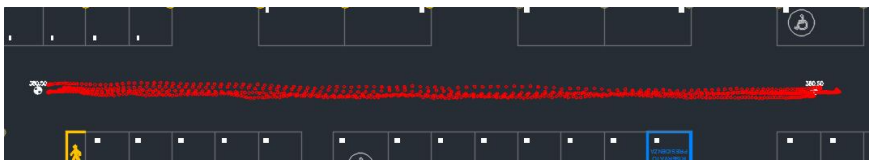


FIGURE 5.28: In red, all the UWB traces.

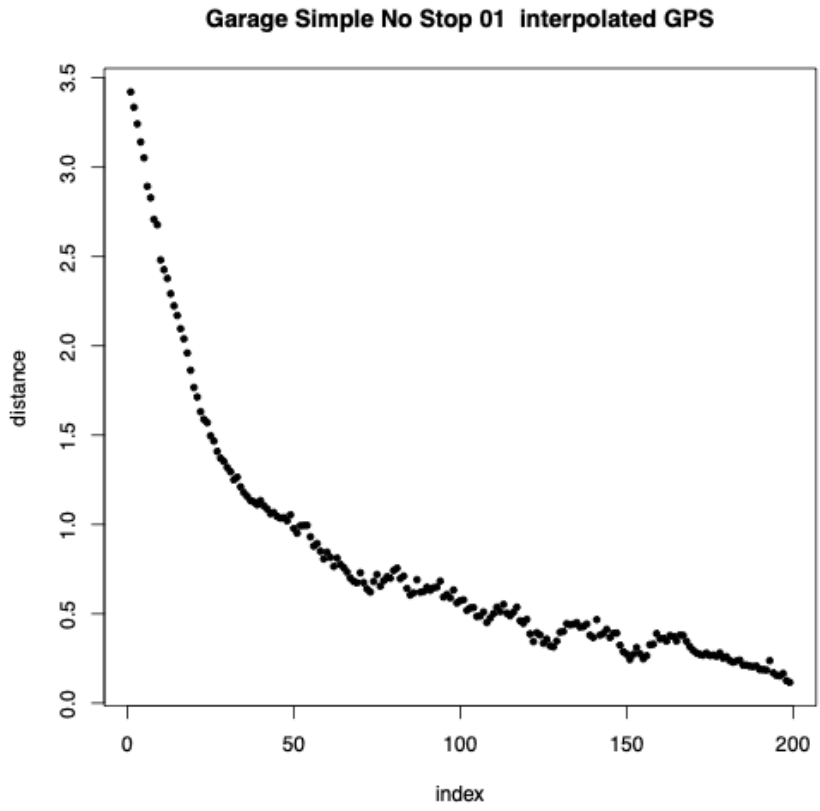


FIGURE 5.29: The difference measurement in meters over time between the V2X trace and the UWB, applying the index that minimizes the average difference.

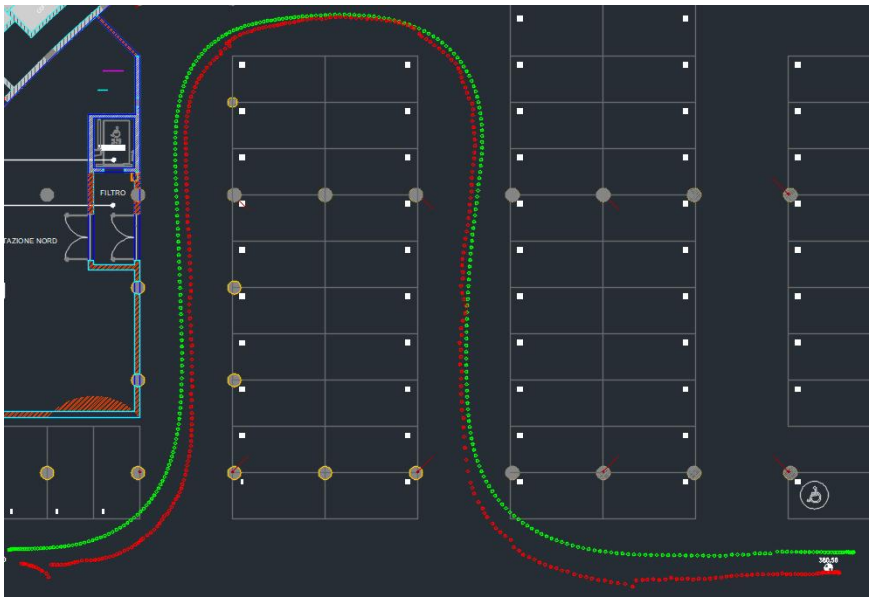


FIGURE 5.30: Single trace of the curve path with V2X in green and UWB in red.

FIGURE 5.31: In green, all the V2X traces.

FIGURE 5.32: In red, all the UWB traces.

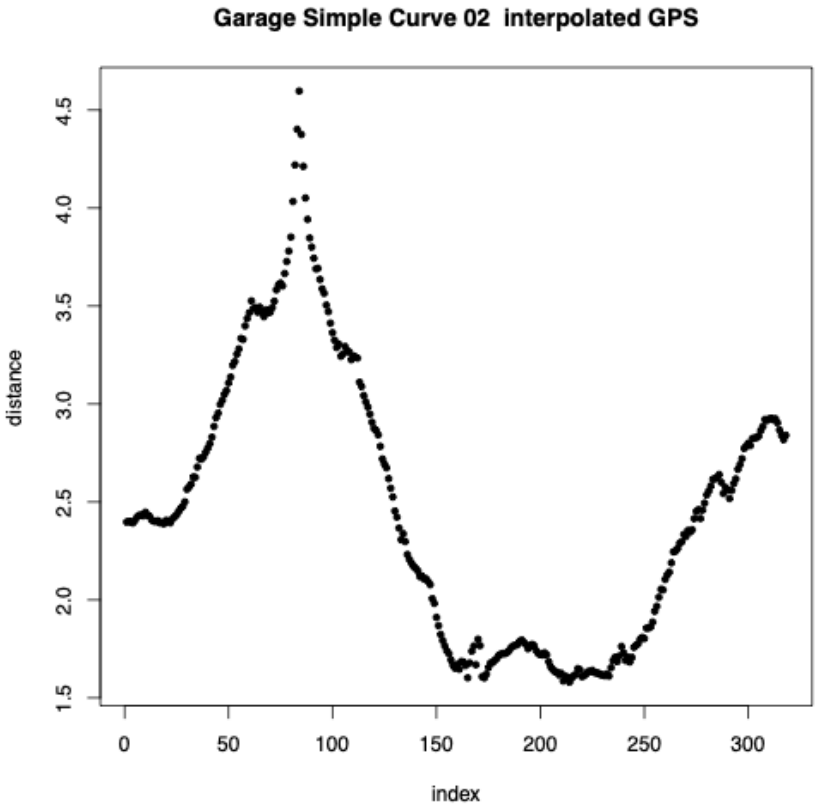


FIGURE 5.33: The difference in meters between the V2X and UWB over a single trace with the index that minimizes the average difference.

TABLE 5.6: Index offset and error for the indoor Simple Path with curve.

	Sampling rate 10Hz		Artificially increased sampling rate			
Trial Index (No.)	Index Offset (No. of samples)	Mean Distance (m)	Index offset (No. of samples) minimizing mean distance	Mean Distance (m)	Index offset -3	Mean Distance (m)
2	1	2.466	5	2.465	-31	2.554
3	-1	0.967	-9	0.966	-31	1.105
4	4	3.338	43	3.33	-31	3.570
5	0	1.069	4	1.066	-31	1.261

As before, we calculate the index offset that minimizes the average distance between each pair of traces for both the original 10Hz data, the artificially upsampled V2X data, and with the index offset of -31. In all cases, the average distance is greater than 1 m.

By evaluating the distance over time in Figure 5.33, we first note that the Y-axis showing the distance has a minimum of 1.5 m, meaning that the difference between the traces was never less than 1.5 m. We see that initially the difference grows, then over time improves, then deteriorates again at the end. The best behavior is actually around the curve of the parking lot, while the straight portions and the moments when the car is driving around the curve are the worst. Even if the UWB system had visually reasonable performance during the experiments, some trials highlight minor problems. For instance, Figure 5.34 shows a deviation of the data present in 2 of the 5 repetitions. This problem likely arises either due to the multipath or from a poor anchor selection for calculating the position. We presume that increasing the number of anchors would reduce the problem. Notably, this deviation is present at the end of the experiment, demonstrating that the other issues shown in the previous section were correctly mitigated by changing the anchor configuration.

5.4 Concluding Remarks

The table below offers a summary of the data presented above. For each experiment, we present the average, minimum, and maximum difference values for the index offset shown in the final column.

TABLE 5.7: Index offset and error for the indoor Simple Path without Stop.

Experiment		**Average Difference (m)**	**Minimum Difference (m)**	**Maximum Difference (m)**	**index offset**
Outdoor: GPS/UWB	8-shape	0.098	0.092	0.105	-34
	Circle	0.082	0.079	0.086	-31
Garage: V2X/UWB	Simple	2.773	1.167	3.869	-31
	Curve	2.122	1.105	3.570	-31
	Parking	3.077	1.840	4.897	-31
	Wave	4.529	2.145	10.378	-31

It can clearly be seen from the outdoor experiments with GPS and UWB that the UWB system produces locations with a small difference w.r.t. GPS, less than 10 cm.

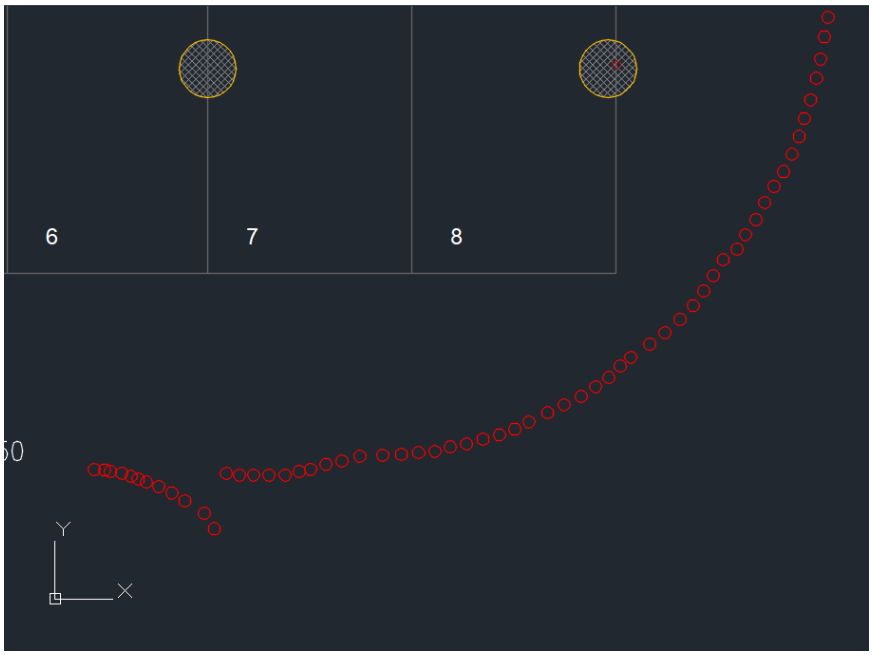


FIGURE 5.34: An anomalous area of the UWB data trace.

As detailed above, the threats to accuracy arise when the vehicle is near the edge of the covered area or when there are obstacles near the path such as other parked vehicles. The average values presented in this summary are shown compared to an up-sampled GPS trace, minimizing the error arising from the unsynchronized sampling of the two systems.

For the experiments in the garage, where UWB is compared to the commercial V2X system, we see that the difference between the measurements of the two systems increases dramatically, on average 2.2 to 4.5 m, but in some trials up to 10 m. As the ground truth is difficult to obtain in the garage environment for a moving vehicle, our work presented above demonstrates that the UWB system as installed produces errors no greater than 30 cm, and often much less. Therefore, we have confidence that it provides a reasonable point for comparison. As in the outdoor trials, we chose to up-sample the V2X trace before comparing it to the UWB trace. Further, we show results in this summary for the offset of -31, which corresponds to the offset observed on the terrace, even though this offset does not minimize the difference between the traces. Values for this minimization can be seen in the document above. Analysis of the V2X traces show large variation across the trials, but no systematic error is observed. For example, we did not observe that the first trial was better than later trials, nor did we see a consistent offset between the traces. In general, based on our experiments during the setup phase of the UWB system, we observed that the threats to the accuracy of the system are much the same as on the terrace, namely parked cars and obstacles such as concrete pillars. Further, as the area under consideration is less regular than on the terrace, the algorithm used to select the anchors does not always choose "the best" anchors, resulting in larger errors in some areas. Nevertheless, by placing multiple anchors, and carefully orienting their antenna to account for these challenges, the resulting system is quite accurate.

Chapter 6

Conclusions

Proximity Detection is a fundamental task of the IoT paradigm. *"The ability of an IoT device to sense the environment in search of similar devices and to know the distance to them"* represents the beating heart of a plethora of applications: from proximity warning systems to data logging in wildlife environments for research purpose, each application presents different requirements and challenges. Indeed, even if such protocols can seem straightforward to design, we have shown that finding an equilibrium among *accuracy*, *energy consumption* and *latency* is a complex task.

In this thesis, we faced this trade-off during the definition of two novel proximity detection protocols, Janus and Hermes, each of which provides promising results. We have also concretely demonstrated the use of Janus in a real application, offering a complete research study from the data collection to sociological analysis and impact. Serving as the backbone of this thesis, the UWB radio has been successfully used in both protocols, demonstrating its applicability in proximity detection. The quality of the UWB ranging has also been illustrated even in harsh environment in Chapter 5, where the radio showed to be reliable in presence of strong multipath components.

More in detail, with Janus we have shown that the use of a dual-radio architecture, with BLE and UWB allowed the development of a protocol that is both accurate and energy efficient. We illustrated how the two radios can be used in synergy, using BLE for continuous neighbor discovery and to coordinate the UWB radio that is exploited only for ranging on-demand. We demonstrated the reliability of Janus with controlled experiments and confirmed our results with a mathematical model. Moreover, we tested the protocol in a motion capture facility, demonstrating its accuracy, with mean ranging errors of approximately 30 cm. Furthermore, the energy efficiency of the protocol has been shown, illustrating that Janus can achieve weeks of operation depending on the configuration.

Janus was then at the center of a complete research study in which we used the protocol in the context of measuring close proximity interactions among children at camps during the summer 2020, when the COVID-19 pandemic was still an international focus. Here the combination of accuracy and energy efficiency was crucial to the successful application of the protocol. In this experience, our collaboration with data analysis researchers led to an in-depth analysis of the data, enabled also by the accurate measurements collected from the devices. The main target of this study was to understand the dynamics of social interactions during the pandemic and their possible effect on the spread of the disease. The collected data allowed us to derive several key insights into the duration and proximity patterns characterizing the child-child and the educator-child interactions. Interestingly, the precise data enabled by our device, allowed us to define accurate risk levels, that together with meta-data such as the type of activity being performed, allowed us to understand fundamental information about the proximity patterns.

While we successfully demonstrated the positive qualities of Janus we also consider its drawbacks. Specifically, splitting of the neighbor discovery and ranging across two different radios introduced a latency of up to three epochs between discovery and ranging. This problem represented the starting point for Hermes, the second proximity detection protocol presented in this thesis. We have illustrated that, by using only the UWB radio, it is possible to design a low-latency protocol while still maintaining reasonably energy consumption for IoT devices. We have

demonstrated this through a custom simulator, a flexible tool that, by simulating the discovery between the nodes, showed the reliability and low-latency enabled by Hermes. Indeed, we have seen that, even with 15 nodes in the same area, it is possible to discover 96% of them within 2 epochs. Moreover, we have implemented the protocol on real devices and demonstrated that the results achieved with the simulator are close to what we see in practice. Finally, we illustrated two methods for increasing the energy savings, deep sleep and preamble sniffing, showing their benefits as well as drawbacks.

Overall this thesis provided a complete exploration for proximity detection and the aforementioned challenges it poses. However, the work is continuing and we have several future goals. In particular, in Hermes we are developing a mathematical model, to make our work more robust and verify the simulation and implementation results. Further, we plan to realize an extensive experiments campaign in the CLOVES [69] testbed, to evaluate both discovery and ranging of the protocol. In fact, in the near future we plan to demonstrate that also Hermes allows high ranging accuracy. Moreover, we will plan to exploit Hermes for a novel research study, moving beyond our laboratory to have real world impact with our research results.

Bibliography

[1] M. Abbas et al. "WiDeep: WiFi-based Accurate and Robust Indoor Localization System using Deep Learning". In: *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (2019).

[2] N. Aharony et al. "Social fMRI: Investigating and shaping social mechanisms in the real world". In: *Pervasive and Mobile Computing* 7 (2011).

[3] N. Ahmed et al. "A Survey of COVID-19 Contact Tracing Apps". In: *IEEE Access* 8 (2020).

[4] A. Alshamsi et al. "Beyond contagion: Reality mining reveals complex patterns of social influence". In: *PloS One* 10 (8 2015), e0135740.

[5] Md S. Aman et al. "Reliability evaluation of iBeacon for micro-localization". In: *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (2016).

[6] *Apple iBeacon*. URL: https://developer.apple.com/ibeacon/.

[7] I. Awolusi, E. Marks, and M. Hallowell. "Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices". In: *Automation in Construction* 85 (2018).

[8] C. Beaulieu. "Intercultural Study of Personal Space: A Case Study". In: *Journal of Applied Social Psychology* 34.4 (2004).

[9] L. Bioglio et al. "Recalibrating disease parameters for increasing realism in modeling epidemics in closed settings". In: *BMC infectious diseases* 16.1 (2016), pp. 1–15.

[10] A. Biri et al. "SociTrack: Infrastructure-Free Interaction Tracking through Mobile Sensor Networks". In: *Proc. of MobiCom*. 2020.

[11] P. Block et al. "Social network-based distancing strategies to flatten the COVID-19 curve in a post-lockdown world". In: *Nature Human Behaviour* 4.6 (2020), pp. 588–596.

[12] G. Brankston et al. "Transmission of influenza A in human beings". In: *The Lancet Infectious Diseases* 7 (5 2007), pp. 257–265.

[13] J. M. Brauner et al. "Inferring the effectiveness of government interventions against COVID-19". In: *Science* 371 (2021), eabd9338.

[14] J. T. Brooks and J. C. Butler. "Effectiveness of mask wearing to control community spread of SARS-CoV-2". In: *JAMA Insights* 325.10 (2021), pp. 998–999.

[15] T. C. Bulfone et al. "Outdoor transmission of SARS-CoV-2 and other respiratory viruses: A systematic review". In: *The Journal of Infectious Diseases* 223.4 (2021), pp. 550–561.

[16] *Bump*. URL: https://www.bump-space.com/.

[17] Y. Cao, A. Dhekne, and M. Ammar. "6Fit-A-Part: A Protocol for Physical Distancing on a Custom Wearable Device". In: *Proc. of ICNP*. 2020.

[18] C. Cattuto et al. "Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks". In: *PLOS ONE* 5 (2010).

[19] G. Cencetti et al. "Digital proximity tracing on empirical contact networks for pandemic control". In: *Nature Communications* 12.1 (2021), pp. 1–12.

[20] H.-Y. Cheng et al. "Contact tracing assessment of COVID-19 transmission dynamics in Taiwan and risk at different exposure periods before and after symptom onset". In: *JAMA Internal Medicine* 180.90 (2020), pp. 1156–1163.

[21] T. Choudhury and A. Pentland. "Sensing and modeling human networks using the sociometer". In: *Proc. of the 7th IEEE Int. Symp. on Wearable Computers*. White Plains, NY, USA, 2003, pp. 216–222.

[22] D. K Chu et al. "Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis". In: *The Lancet* 395.10242 (2020), pp. 1973–1987.

[23] M. Collotta et al. "Bluetooth 5: A Concrete Step Forward toward the IoT". In: *IEEE Communications Magazine* 56.7 (July 2018), pp. 125–131.

[24] P. Corbalán and G.P. Picco. "Ultra-wideband Concurrent Ranging". In: *ACM Trans. on Sensor Networks* 16.4 (Oct. 2020), pp. 1–41.

[25] Decawave. *DWM1001-System Overview*. 2018. URL: https://www.qorvo.com/products/d/da007974.

[26] I. Dotlic, A. Connell, and M. McLaughlin. "Ranging Methods Utilizing Carrier Frequency Offset Estimation". In: *Proc. of the 15th Workshop on Positioning, Navigation and Communications (WPNC)*. 2018.

[27] P. Dutta and D. Culler. "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications". In: (2008), 71–84.

[28] A. Duval et al. "Close proximity interactions support transmission of ESBL-*K. pneumoniae* but not ESBL-*E. coli* in healthcare settings". In: *PloS Computational Biology* 15 (5 2019), e1006496.

[29] A. Duval et al. "Measuring dynamic social contacts in a rehabilitation hospital: effect of wards, patient and staff characteristics". In: *Scientific Reports* 8 (2018), p. 1686.

[30] European Centre for Disease Prevention and Control. *Contact tracing: Public health management of persons, including healthcare workers, having had contact with COVID-19 cases in the European Union*. https://www.ecdc.europa.eu/sites/default/files/documents/covid-19-public-health-management-contact-novel-coronavirus-cases-EU.pdf.

[31] European Centre for Disease Prevention and Control. *Resource estimation for contact tracing, quarantine and monitoring activities for COVID-19 cases in the EU/EEA*. https://www.ecdc.europa.eu/en/publications-data/resource-estimation-contact-tracing-quarantine-and-monitoring-activities-covid-19.

[32] K. Farrahi, R. Emonet, and M. Cebrian. "Epidemic contact tracing via communication traces". In: *PloS one* 9.5 (2014), e95133.

[33] D. M. Feehan and A. S. Mahmud. "Quantifying population contact patterns in the United States during the COVID-19 pandemic". In: *Nature Communications* 12.1 (2021), pp. 1–9.

[34] J. Fournet and A. Barrat. "Estimating the epidemic risk using non-uniformly sampled contact data". In: *Scientific reports* 7.1 (2017), pp. 1–14.

[35] S. Funk, M. Salathé, and V. A. A. Jansen. "Modelling the influence of human behaviour on the spread of infectious diseases: A review". In: *Journal of the Royal Society Interface* 7 (50 2010).

[36] M. Génois et al. "Compensating for population sampling in simulations of epidemic spread on temporal contact networks". In: *Nature communications* 6.1 (2015), pp. 1–13.

[37] C. Gomez, J. Oller, and J. Paradells. "Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology". In: *Sensors (Basel)* 12.9 (2012), pp. 11734–11753.

[38] *Google Eddystone*. URL: https://github.com/google/eddystone/.

[39] *Google/Apple Exposure Notification (GAEN)*. URL: https://covid19.apple.com/contacttracing/.

[40] S. Greenberg et al. "Proxemic Interactions: The New Ubicomp?" In: *ACM Interactions* 18.1 (2011), pp. 42–50.

[41] T. Greenhalgh et al. "Ten scientific reasons in support of airborne transmission of SARS-CoV-2". In: *The Lancet* 397.10285 (2021), pp. 1603–1605.

[42] N. Haug et al. "Ranking the effectiveness of worldwide COVID-19 government interventions". In: *Nature Human Behaviour* 4.12 (2020), pp. 1303–1312.

[43] V.S. Hertzberg et al. "Contact networks in the emergency department: Effects of time, environment, patient characteristics, and staff role". In: *Social Networks* 48 (2017), pp. 181–191.

[44] Hexagon, NovAtel. *Real-Time Kinematic (RTK)*. [Last Access: (14-03-2023)]. 2015. URL: https://novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk.

[45] S. Hsiang et al. "The effect of large-scale anti-contagion policies on the COVID-19 pandemic". In: *Nature* 584 (2020), pp. 262–267.

[46] C. Huang et al. "Insights into the transmission of respiratory infectious diseases through empirical human contact networks". In: *Scientific Reports* 6 (3184 2016).

[47] W. Huang et al. "Opo: A Wearable Sensor for Capturing High-Fidelity Face-to-Face Interactions". In: *Proc. of SenSys*. 2014.

[48] Timofei I. et al. "Janus: Dual-radio Accurate and Energy-Efficient Proximity Detection". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5.4 (Dec. 2021).

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

[50] L. Isella et al. "Close encounters in a pediatric ward: Measuring face-to-face proximity and mixing patterns with wearable sensors". In: *PloS One* 6 (2 2011), e17144.

[51] C. I. Jarvis et al. "Quantifying the impact of physical distance measures on the transmission of COVID-19 in the UK". In: *BMC Medicine* 18 (2020), p. 124.

[52] N. R Jones et al. "Two metres or one: what is the evidence for physical distancing in covid-19?" In: *BMJ* 370 (2020).

[53] C. Julien et al. "BLEnd: Practical Continuous Neighbor Discovery for Bluetooth Low Energy". In: *Proc. of IPSN*. 2017.

[54] M. Kendall et al. "Epidemiological changes on the Isle of Wight after the launch of the NHS Test and Trace programme: a preliminary analysis". In: *The Lancet Digital Health* 2.12 (2020), e658–e666.

[55] J. Kolakowski et al. "UWB/BLE Tracking System for Elderly People Monitoring". In: *Sensors* 20 (2020).

[56] F.-Y. Lan et al. "Work-related COVID-19 transmission in six Asian countries/areas: A follow-up study". In: *PloS One* 15.5 (2020), e0233588.

[57] Q. J. Leclerc et al. "What settings have been linked to SARS-CoV-2 transmission clusters?" In: *Wellcome Open Research* 5.83 (2020).

[58] D.J. Leith and S. Farrell. "Measurement-based evaluation of Google/Apple Exposure Notification API for proximity detection in a light-rail tram". In: *PLOS ONE* 15 (9 2020).

[59] Trystan Leng et al. “The effectiveness of social bubbles as part of a COVID-19 lockdown exit strategy, a modelling study”. In: *Wellcome Open Research* 5.213 (2020), p. 213.

[60] Leoni, E. et al. “Measuring close proximity interactions in summer camps during the COVID-19 pandemic”. In: *EPJ Data Sci.* 11.1 (2022).

[61] N. H. L. Leung. “Transmissibility and transmission of respiratory viruses”. In: *Nature Reviews Microbiology* (2021).

[62] Y. Li et al. “Face masks to prevent transmission of COVID-19: A systematic review and meta-analysis”. In: *American Journal of Infection Control* (2020).

[63] Decawave Ltd. *DW1000 Device Driver Application Programming Interface (API) Guide*. Version 2.7. 2016.

[64] Decawave Ltd. *DWM1001—Datasheet*. 2017. URL: https://www.decawave.com/dwm1001/datasheet/.

[65] C. Martella et al. “Leveraging Proximity Sensing to Mine the Behavior of Museum Visitors”. In: *Proc. of PerCom*. 2016.

[66] R. Mastrandrea et al. “Enhancing the evaluation of pathogen transmission risk in a hospital by merging hand-hygiene compliance and contact data: a proof-of-concept study”. In: *BMC research notes* 8.1 (2015), pp. 1–11.

[67] M. J. McGlynn and S. A. Borbash. “Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks”. In: (2001), 137–145.

[68] Vu Anh Minh Le et al. “Human Occlusion in Ultra-wideband Ranging: What Can the Radio Do for You?” In: (2022).

[69] D. Molteni et al. “Cloves: A Large-Scale Ultra-Wideband Testbed”. In: SenSys ’22 (2023).

[70] A. Montanari et al. “A Study of Bluetooth Low Energy performance for human proximity detection in the workplace”. In: *Proc of PerCom*. 2017.

[71] L Morawska and D K Milton. “It is time to address airborne transmission of coronavirus disease 2019 (COVID-19)”. In: *Clinical Infectious Diseases* 71 (2020), 2311—2313.

[72] J.A. Moreno López et al. “Anatomy of digital contact tracing: Role of age, transmission setting, adoption, and case detection”. In: *Science Advances* 7.15 (2021), eabd8750.

[73] D. Neirynck, E. Luk, and M. McLaughlin. “An Alternative Double-Sided Two-Way Ranging Method”. In: *Proc. of the 13th IEEE Workshop on Positioning, Navigation and Communications (WPNC)*. 2016.

[74] D. O. Olguín et al. “Sensible organizations: Technology and methodology for automatically measuring organizational behavior”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.1 (2009), pp. 43–55.

[75] T. Obadia et al. “Detailed contact data and the dissemination of Staphylococcus aureus in hospitals”. In: *PLoS computational biology* 11.3 (2015), e1004170.

[76] T. Otim et al. “Effects of the Body Wearable Sensor Position on the UWB Localization Accuracy”. In: *Electronics* (2019).

[77] L. Ozella et al. “Close encounters between infants and household members measured through wearable proximity sensors”. In: *PLOS ONE* 13 (2018).

[78] F. Palumbo et al. “A stigmergic approach to indoor localization using Bluetooth Low Energy beacons”. In: *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2015).

[79] G.P. Picco et al. “Geo-referenced Proximity Detection of Wildlife with WildScope: Design and Characterization”. In: *Proc. of IPSN*. 2015.

[80] Tommaso Polonelli, Simon Schläpfer, and Michele Magno. "Performance Comparison between Decawave DW1000 and DW3000 in low-power double side ranging applications". In: (2022), pp. 1–6.

[81] *ProDongle*. URL: https://www.prodongle.com/en/solution/social-distancing/.

[82] QORVO. *DWM1000-Datasheet*. 2022. URL: https://www.qorvo.com/products/d/da007946.

[83] QORVO. *DWM1000-User Manual*. 2022. URL: https://www.qorvo.com/products/d/da007967.

[84] QORVO. *DWM1001-DEV*. 2022. URL: https://www.qorvo.com/products/p/DWM1001-DEV.

[85] QORVO. *DWM3000*. 2022. URL: https://www.qorvo.com/products/p/DWM3000.

[86] B. Rader et al. "Mask-wearing and control of SARS-CoV-2 transmission in the USA: A cross-sectional study". In: *The Lancet Digital Health* 3.3 (2021), E148–E157.

[87] J. M. Read, K. T. Eames, and W. J. Edmunds. "Dynamic social networks and the implications for the spread of infectious disease". In: *Journal of The Royal Society Interface* 5 (2008), pp. 1001–1007.

[88] T. Renzler et al. "Improving the Efficiency and Responsiveness of Smart Objects Using Adaptive BLE Device Discovery". In: *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*. SMARTOBJECTS '18. 2018.

[89] P. Rodríguez et al. "A population-based controlled experiment assessing the epidemiological impact of digital contact tracing". In: *Nature Communications* 12.587 (2021).

[90] B.R. Rowe et al. "Simple quantitative assessment of the outdoor versus indoor airborne transmission of viruses and COVID-19". In: *Environmental Research* 198 (2021), p. 111189.

[91] M. Salathé et al. "Early evidence of effectiveness of digital contact tracing for SARS-CoV-2 in Switzerland". In: *Swiss Medical Weekly* 150 (2020), w20457.

[92] Marcel Salathé et al. "A high-resolution human contact network for infectious disease transmission". In: *Proceedings of the National Academy of Sciences* 107.51 (2010), pp. 22020–22025.

[93] J. Staiano et al. "Friends Don't Lie: Inferring Personality Traits from Social Network Structure". In: *Proc. of UbiComp*. 2012.

[94] J. Stehlé et al. "High-resolution measurements of face-to-face contact patterns in a primary school". In: *PloS One* 6 (8 2011), e23176.

[95] J. Stehlé et al. "Simulation of an SEIR infectious disease model on the dynamic contact network of conference attendees". In: *BMC Medicine* 9 (1 2011), p. 87.

[96] B. Tao et al. "An RFID-Based Mobile Robot Localization Method Combining Phase Difference and Readability". In: *IEEE Transactions on Automation Science and Engineering* (2021).

[97] *Ubudu*. URL: https://www.ubudu.com/social-distancing-assistant-for-enterprises-to-provide-a-safe-workplace-until-covid-19-goes-away/.

[98] P. Vanhems et al. "Estimating potential infection transmission routes in hospital wards using wearable proximity sensors". In: *PloS One* 8 (9 2013), e73970.

[99] D. Vecchia et al. “Playing with Fire: Exploring Concurrent Transmissions in Ultra-wideband Radios”. In: *Proc. of SECON*. 2019.

[100] K. Wang, Xufei Mao, and Yunhao Liu. “BlindDate: A Neighbor Discovery Protocol”. In: (2013), pp. 120–129.

[101] R. Want, B. N. Schilit, and S. Jenson. “Enabling the Internet of Things”. In: *IEEE Computer* 48.1 (Jan. 2015), pp. 28–35.

[102] R. A. Weinstein et al. “Transmission of influenza: Implications for control in health care settings”. In: *Clinical Infectious Diseases* 37 (8 2003), pp. 1094–1101.

[103] Wikipedia. *List Smartphone with UWB*. 2022. URL: https://en.wikipedia.org/wiki/List_of_UWB-enabled_mobile_devices.

[104] C.C. Wilmers et al. “The golden age of bio-logging: how animal-borne sensors are advancing the frontiers of ecology”. In: *Ecology* 96 (2015).

[105] C. Wymant et al. “The epidemiological impact of the NHS COVID-19 app”. In: *Nature* (2021).

[106] Y. Yoshimura, A. Krebs, and C. Ratti. “Noninvasive Bluetooth Monitoring of Visitors’ Length of Stay at the Louvre”. In: *IEEE Pervasive Computing* 16.2 (Mar. 2017).

[107] 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.

[108] J. Zhang et al. “Changes in contact patterns shape the dynamics of the COVID-19 outbreak in China”. In: *Science* 368 (2020), pp. 1481–1486.

[109] J. Zhang et al. “The impact of relaxing interventions on human contact patterns and SARS-CoV-2 transmission in China”. In: *Science Advances* 368.19 (2021), eabe2584.

[110] Y. Zhuang et al. “Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons”. In: *Sensors* 16 (2016).