Janus: Dual-radio Accurate *and* Energy-Efficient Proximity Detection

TIMOFEI ISTOMIN, University of Trento, Italy
ELIA LEONI, Bruno Kessler Foundation, Italy
DAVIDE MOLTENI, University of Trento, Italy
AMY L. MURPHY, Bruno Kessler Foundation, Italy
GIAN PIETRO PICCO, University of Trento, Italy
MAURIZIO GRIVA, Santer Reply, Italy

Proximity detection is at the core of several mobile and ubiquitous computing applications. These include reactive use cases, e.g., alerting individuals of hazards or interaction opportunities, and others concerned only with logging proximity data, e.g., for offline analysis and modeling. Common approaches rely on Bluetooth Low Energy (BLE) or ultra-wideband (UWB) radios. Nevertheless, these strike opposite tradeoffs between the accuracy of distance estimates quantifying proximity and the energy efficiency affecting system lifetime, effectively forcing a choice between the two and ultimately constraining applicability.

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.

CCS Concepts: • **Networks** → **Network protocol design**; • **Human-centered computing** → **Ubiquitous and mobile computing**.

Additional Key Words and Phrases: proximity detection, ultra-wideband (UWB), Bluetooth

ACM Reference Format:
Timofei Istomin, Elia Leoni, Davide Molteni, Amy L. Murphy, Gian Pietro Picco, and Maurizio Griva. 2021. Janus: Dual-radio Accurate *and* Energy-Efficient Proximity Detection. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 162 (December 2021), 33 pages. https://doi.org/10.1145/3494978

1 INTRODUCTION

Proximity detection is a fundamental building block of several mobile and ubiquitous computing 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 [1] and Google Eddystone [3], are typically deployed in designated places of interest, e.g., the exhibits of a museum,

Authors' addresses: Timofei Istomin, University of Trento, Trento, Italy, timofei.istomin@unitn.it; Elia Leoni, Bruno Kessler Foundation, Trento, Italy, eleoni@fbk.eu; Davide Molteni, University of Trento, Trento, Italy, davide.molteni@unitn.it; Amy L. Murphy, Bruno Kessler Foundation, Trento, Italy, murphy@fbk.eu; Gian Pietro Picco, University of Trento, Trento, Italy, gianpietro.picco@unitn.it; Maurizio Griva, Santer Reply, Turin, Italy.

2474-9567/2021/12-ART162 $15.00
https://doi.org/10.1145/3494978

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 [38]. *Proximity warning systems*, which typically rely on different technologies [13], 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.

In several other contexts, proximity detection is instead exploited to *acquire data, enabling its analysis*. This is actually another typical use of beacons that, by logging user proximity events, enable analysis and modeling of the 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* [23], the study of the space individuals set between themselves and others, and many others where quantifying *social contact* is key, as in studying relationships [34], or the social behavior of people at scale [36]. Similarly, *biologging*, a recent trend in biology, focuses on proximity detection among animals to understand their interactions and behavior [39].

Interestingly, the recent COVID-19 pandemic fueled a demand for systems along both perspectives above. The use of personal devices such as smartphones [12] or dedicated "proximity tags" [2] 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.

1.1 State-of-the-art Dilemma: Accurate or Energy-efficient?

In the applications above, the accuracy of distance estimation is clearly a key metric that directly impacts the higher-level use of proximity information, e.g., to determine the threshold at which to trigger an alarm, or correctly inform the analysis of contact data. In turn, the quality of this metric directly descends from the specific technology employed. Many are available, significantly overlapping with those commonly used in localization [41]. Here, we focus on RF-based ones as they are by far the most commonly used due to their ubiquity and the ability to double as a communication means.

Bluetooth Low Energy (BLE) devices are a common choice, due to BLE's pervasiveness on personal and wearable devices and other common objects. RFID is another popular choice [18, 34]; in particular, passive tags do not require a power supply enabling very cheap, disposable designs. These technologies have been quite successful in showing the potential of proximity detection for the above classes of applications. Nevertheless, they do not *directly* measure distance, which is instead inferred via the received signal strength indicator (RSSI). This technique typically yields only coarse estimates with errors of meters [42] severely limiting the potential applicability. For instance, in museums their application is usually limited to detecting user presence at the room level [31, 40] 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 [28]. Further, these social studies typically require sub-meter accuracy [15], which exacerbates the problems above.

A promising alternative is offered by ultra-wideband (UWB) radios, 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 WiFi and 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 an infrastructure of fixed reference nodes (anchors), implicitly delimiting the area where detection can occur, posing conflicting tradeoffs 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.

On the other hand, UWB systems can also be used in an infrastructure-less, "peer-to-peer" modality where nodes range against each other. Nevertheless, the highly dynamic environments outlined above require *continuous* monitoring for the presence of neighboring devices that, once discovered, become the target of ranging exchanges. 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 we use in this paper consumes ~80 mA in TX and ~120 mA in RX; the BLE chip on the dual-radio module we exploit consumes only ~4 mA and ~6 mA, respectively. Continuous device discovery over UWB would therefore be very expensive, and potentially wasteful in scenarios where devices spend a significant amount of time away from targets (e.g., individuals alone or workers far from hazards), ultimately hampering significantly the lifetime of UWB-based proximity tags. Frequent recharging is cumbersome, increases operational costs, and ultimately limits practical applicability.

Therefore, the state of the art in proximity detection presents domain experts with a dilemma: *Gather accurate proximity information or enjoy weeks or even months of uninterrupted operation?*

1.2 Janus: Accurate and Energy-efficient

The goal of this paper is to remove this dilemma by reconciling the two perspectives into *Janus*, a system providing both accurate and energy-efficient proximity detection. 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 [10] 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. Further, the exemplary applications above express very different needs, requiring significantly different balances among these performance concerns. Therefore, 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.

1.3 Goals, Methodology, and Contributions

After elaborating on these requirements (§2) we present the design of Janus (§3), reconciling accuracy with energy efficiency by combining the respective strengths of BLE and UWB in a novel dual-radio protocol. Moreover, an analytical model (§4) 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 (§5) 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 (§6).

We characterize the performance of Janus in our system-centric evaluation (§7), 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 to the extent described here. 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 (§8), 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 (§9) 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 [30] 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 of Janus bring w.r.t. them and related work (§10) before offering brief concluding remarks and an outlook on future work (§11).

2 REQUIREMENTS, ASSUMPTIONS, AND DESIGN GOALS

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 (§10). 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 patient near the ward exit door). Our high-level goal 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 [35]. Moreover, the recent Google-Apple exposure Notification (GAEN) [4] 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 (§7) 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 brings 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 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 [41] 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 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.

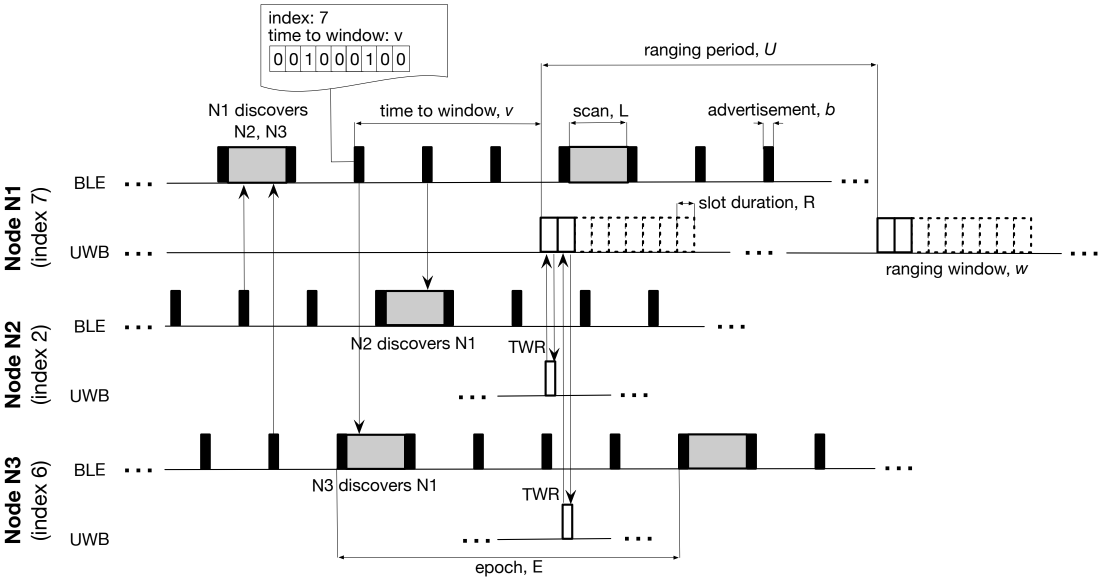


Fig. 1. The Janus protocol. The illustration is a complete account only for node $N1$. Although $N2$ and $N3$ also discover each other during their scans and schedule their own ranging windows, the corresponding portions of the schedule are omitted here for readability.

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 (§4) whose validity we ascertain experimentally (§7).

3 DUAL-RADIO DISCOVERY AND RANGING

Janus merges BLE-based neighbor discovery and UWB ranging into a single energy-efficient protocol (Figure 1) coordinating these two core operations and harmonizing them w.r.t. the requirements above.

3.1 Neighbor Discovery

Janus is built atop BLEnd [25], 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 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.

3.2 Ranging

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 [10] 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 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 [10, 33], 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 [22, 29], is known [21] to improve the quality of ranging without extra messages.

3.3 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 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 (§5).

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; *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 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 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 $\Lambda\ _r$, 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.

3.4 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 [10], 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.

3.5 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 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 (§5) the unused portion of the advertisement payload is $13B$ yielding an address space of 104 bits, large enough to satisfy both concerns.

3.6 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 (§8), 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.

4 MODELING THE SUCCESS OF PROXIMITY DETECTION

As we mentioned (§2), 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 next derive a model estimating the probability of successful detection based on the key user configuration parameters. We later show (§7) that this model is in very good accordance with experimental results and can therefore be used in practice to inform the configuration of Janus.

4.1 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.

Our reliance on BLEnd simplifies matters, as this protocol was expressly designed to ensure predictable performance. The original paper [25] 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 (§7), based on different firmware and hardware w.r.t. the one used for validation in [25], confirm the accuracy of the BLEnd model in estimating P_d. Therefore, the problem reduces to estimating the probability P_r of successful ranging.

4.2 Probability of Successful Ranging

We assume, without detriment to model accuracy (§7), 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 (§3). 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

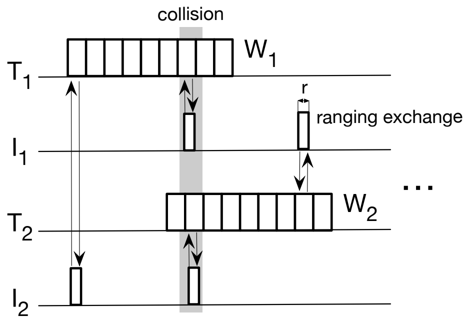


Fig. 2. Modeling Janus: The TWR exchanges of four co-located nodes fail due to a (partial) overlapping, indicated in gray.

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 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

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

5 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 program 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 exploit 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 ∼1 ms leading to synchronization errors of up to 2 ms due to rounding.

We address 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 optimize the energy spent by the UWB radio by placing it in deep sleep mode (∼50 nA) whenever possible. This, however, incurs a relatively long delay (∼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.

6 FROM A PROTOTYPE TO A FULL-FLEDGED SYSTEM

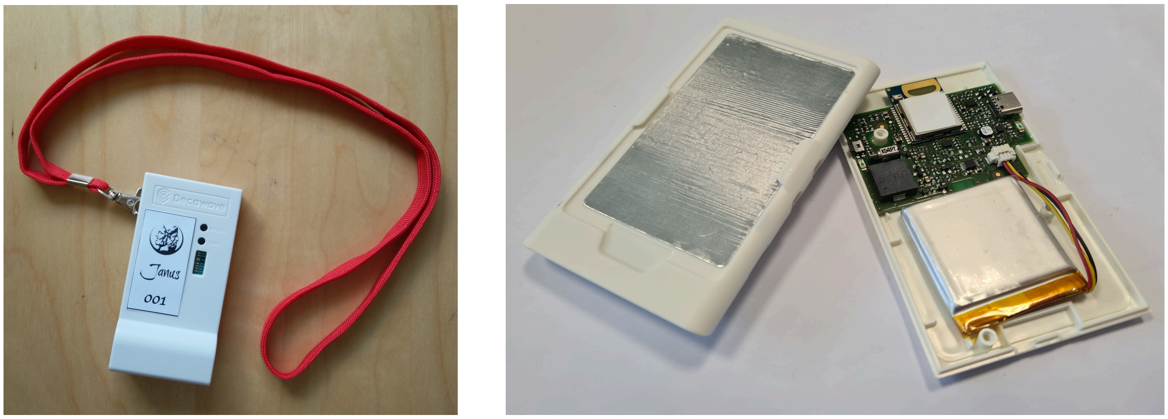
Janus started as a research prototype that we progressively refined to industry-grade level; it is currently integrated in a commercial offer targeting workplace safety. Here we complement the description of Janus, our main focus, with a concise account of other hardware and software components enabling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

6.1 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.

6.2 A Custom Tag

We tested Janus on the Decawave MDEK1001 evaluation kits (Figure 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.



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

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

Fig. 3. Janus nodes used for testing and in commercial deployments.

The current version (Figure 3b) has a badge form factor (106 × 64 × 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.

6.3 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.

7 SYSTEM EVALUATION

We evaluate Janus along different performance dimensions. After discussing our reference configurations (§7.1) we ascertain the latency of first discovery and ranging, and the reliability of detection (§7.2) 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 (§7.3). The representative proximity patterns we experiment with enable us to quantify directly the threats to accuracy posed by antenna orientation and body shielding, elements rarely ascertained in the literature to the extent reported here. Finally, we confirm the energy efficiency of Janus (§7.4) with real measurements, showing it achieves an extended lifetime in configurations of practical interest.

Table 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

7.1 Overall System Configuration and Application Scenarios

We illustrate the key configuration parameters used throughout the evaluation.

7.1.1 Application Requirements and Janus Configuration. Janus is designed to cater for both reactive and logging applications (§2). These are differentiated mainly by their *time bounds* concerned with the latency of first discovery Λ and ranging update rate U, with reactive applications obviously exploiting tighter time bounds.

The Janus configurations we use (Table 1) arise directly from our in-field experience with several real-world applications, including those reported later (§8) 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 (§6) 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 [39], 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 (§9).

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 (§2). 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 1.

7.1.2 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 (§6), as we further discuss when reporting on our real-world experiences with Janus (§8).

7.2 Latency and Reliability

We use a controlled setting to verify whether the target latency of first discovery Λ in our configurations (Table 1) is met by our implementation, and quantify the extra delay incurred by the first ranging. We show that our model of ranging reliability (§4) 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.

7.2.1 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 §7.3.

We explore all configurations in Table 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.

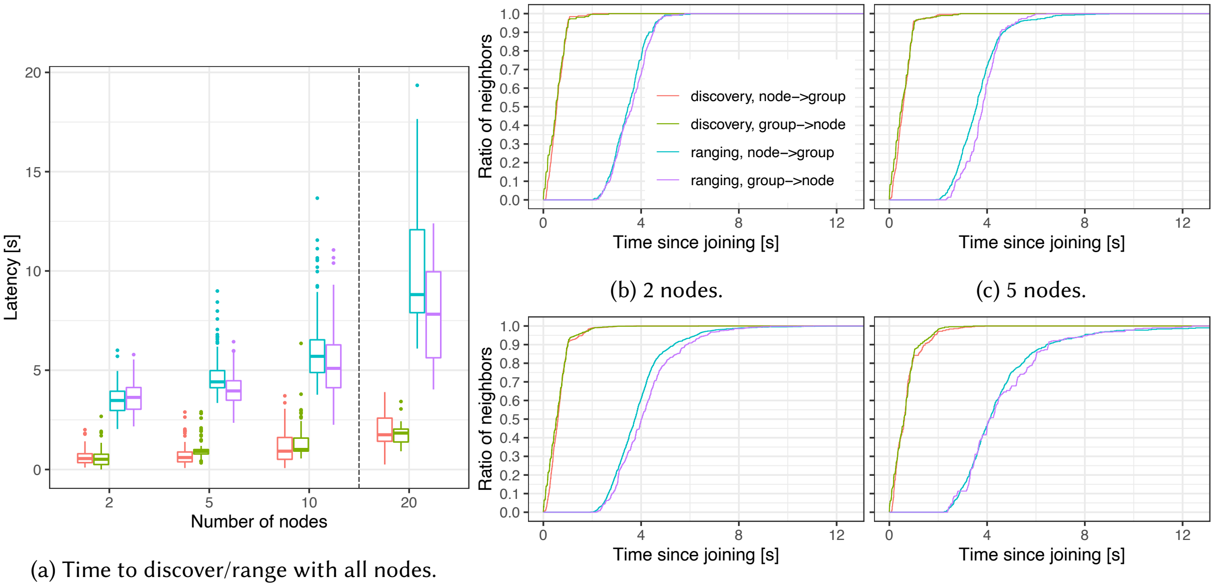
Finally, to avoid biasing the communication schedules of discovery and ranging performed by the two radios, we randomize the node start time within the ranging update interval, U.

7.2.2 Latency. We want to understand the time required by a node approaching a group of other nodes to discover and range with them (node ⟶ group) and vice versa (group ⟶ 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 1 and show results in Figure 4–5. The boxplots (Figure 4a, 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 4b–4e, Figure 5b, 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 [25], 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 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 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 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 (§3) 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



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

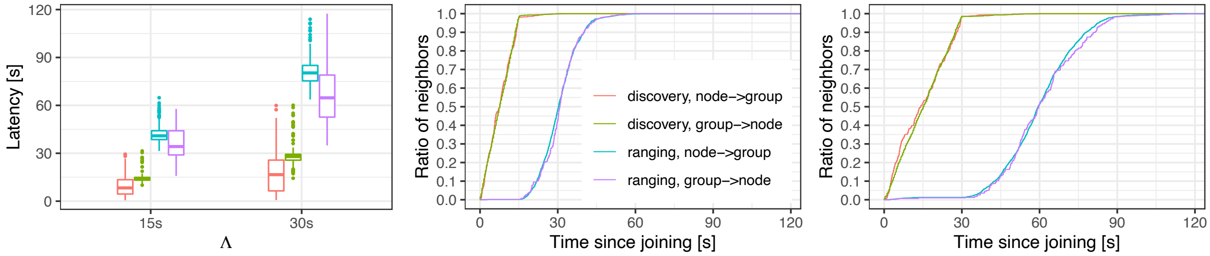
(b) 2 nodes.

(c) 5 nodes.

(d) 10 nodes, expected maximum.

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

Fig. 4. Latency of first discovery and first ranging for the reactive configuration ($\Lambda = 2$ s for 95% of nodes) in Table 1 for different numbers of nodes. These include $N = 10$ (Figure 4d), the expected maximum Janus is configured for, and $N = 20$ (Figure 4e), showing that performance only marginally degrades despite a system scale twice the expected maximum one.



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

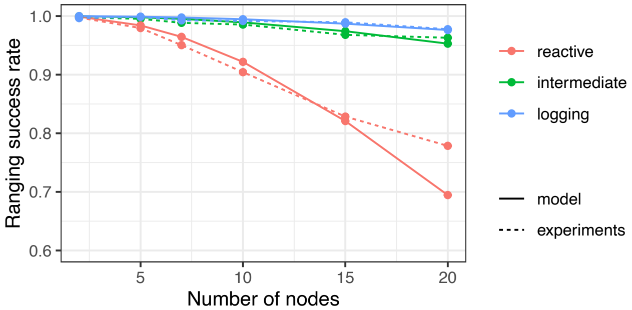
(b) $\Lambda = 15$s

(c) $\Lambda = 30$s

Fig. 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 1 and maximum expected nodes, $N = 10$.

case, this first ranging estimate may become available at some nodes only within 3Λ. On the other hand, the CDFs show that ~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 to range also increases due to failed exchanges induced by collisions among the ranging windows. This is expected and quantified by our model (§4), whose validity we ascertain next.



	Ranging success rate [%]					
	U = 2s		U = 15s		U = 30s	
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

Fig. 6. Janus ranging reliability estimated by the model (§4) and measured in our experiments.

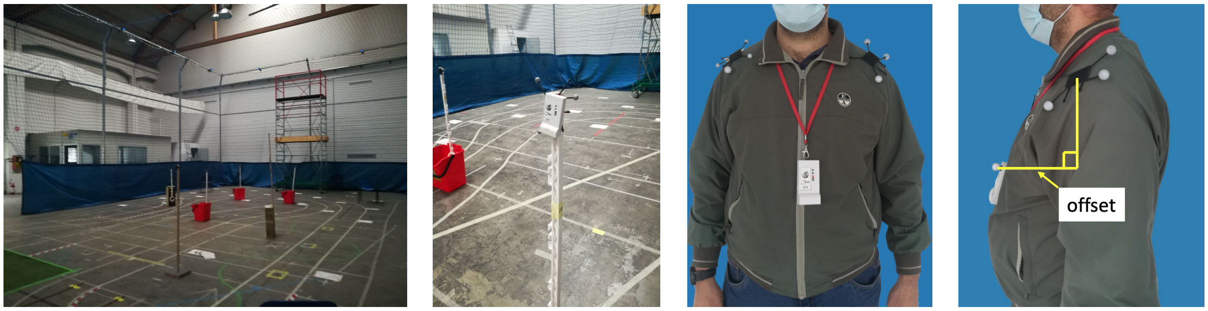
7.2.3 Reliability. Our controlled setup also enables us to accurately estimate the probability that ranging exchanges successfully complete. We use the same three configurations considered thus far, and analyze the impact of different numbers of nodes, both smaller and larger than the maximum expected, $N = 10$. We evaluate the steady-state performance by allowing enough time for nodes to discover each other before collecting metrics. We run each test for a duration of at least $120 \times U$, which effectively enables us 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 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 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 6 also plots the corresponding values of P_r estimated by our model (§4). 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 (~ 8%) from experiments by *under*estimating the probability of success. This is a result of the conservative model assumption that *any* overlap between 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 [37]. 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.

7.3 Ranging Accuracy

We now turn our attention to evaluating 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 is also the opportunity to investigate the effects of antenna orientation and body shielding, which degrade the decimeter-level accuracy commonly associated with UWB, and whose impact is nonetheless rarely ascertained and quantified experimentally to the extent reported here.



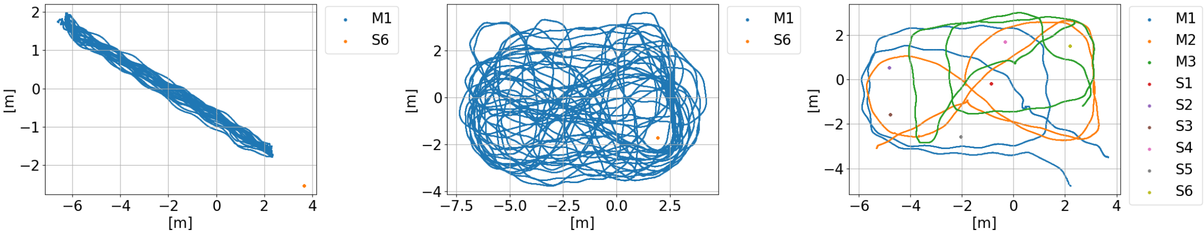
(a) The motion capture area showing six stationary Janus nodes used for compensation calculation and in Multiple.

(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.

Fig. 7. Motion capture setup.



(a) Line.

(b) Random.

(c) Multiple.

Fig. 8. Experimental scenarios and corresponding sample motion traces exported from Optitrack.

7.3.1 Experimental Setup. In all experiments, we used the reactive configuration in Table 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 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). Stationary tags were mounted at chest height on plastic or wooden poles, and equipped with markers (Figure 7b). Mobile tags were instead worn on lanyards around the neck of researchers in our group (Figure 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 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 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 8 with one of the actual Optitrack traces. In Line, our first scenario, a tag is worn by a user moving

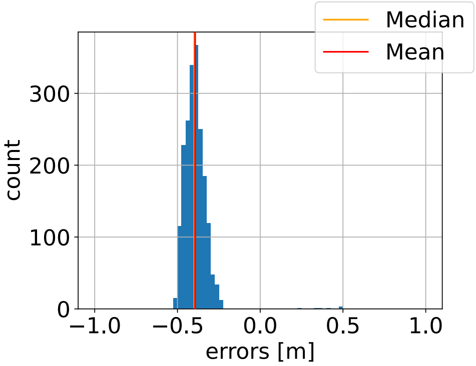


Fig. 9. Distribution of the errors between Janus and OptiTrack distances among 6 fixed nodes. The mean, −39 cm, is applied to all Janus measurement to compensate for arbitrary antenna orientations during testing.

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 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 [16]. 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 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.

7.3.2 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 (§7.2). 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 begin our exploration of the ranging performance with a simple, controlled scenario with a tester walking back and forth. We place one stationary node (S6) near the edge of the area. The tester (M1) begins ~10 m away and walks toward the fixed node, pausing at a distance of ~1.5 m for ~15 s. He then turns around, walks back to the starting position, turns back to face the node, pauses for 15 s, then repeats the process, for a total of 10 minutes (Figure 8a). The experiment was repeated 3 times.

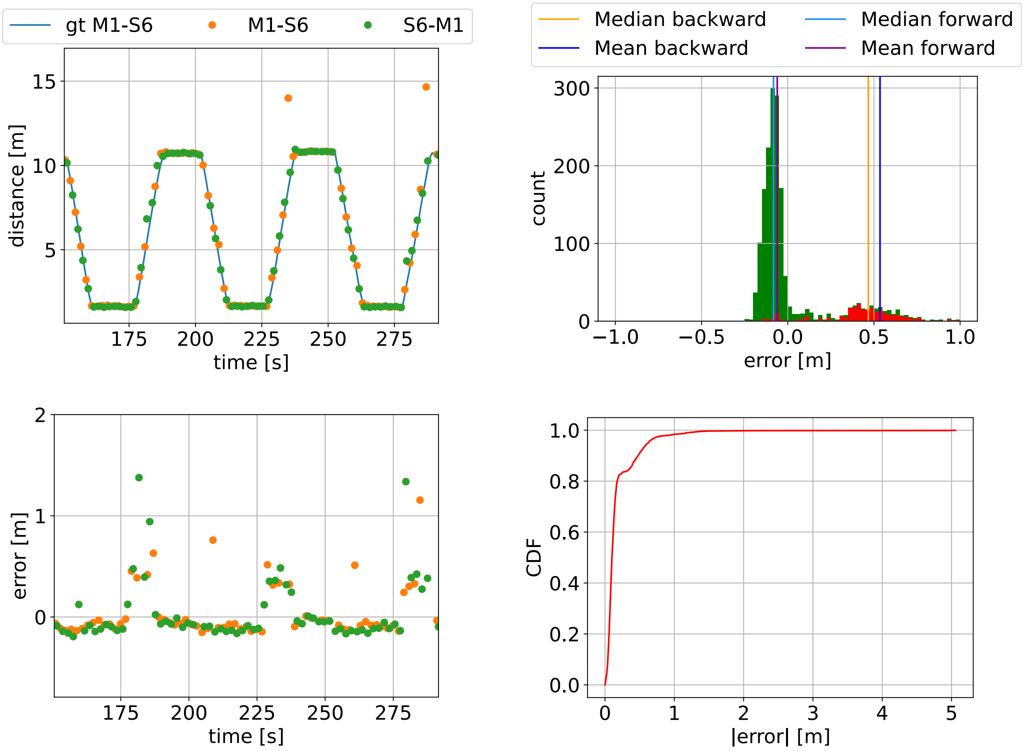
The top of Figure 10a shows part of the trace, with large green and orange dots denoting Janus measurements (e.g., M1–S6 indicates those saved at mobile node M1) and blue points denoting the distances computed from OptiTrack positions, our ground truth; these occur at higher frequency (120 Hz) and appear 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 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 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 (§7.3.1), 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 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

[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).

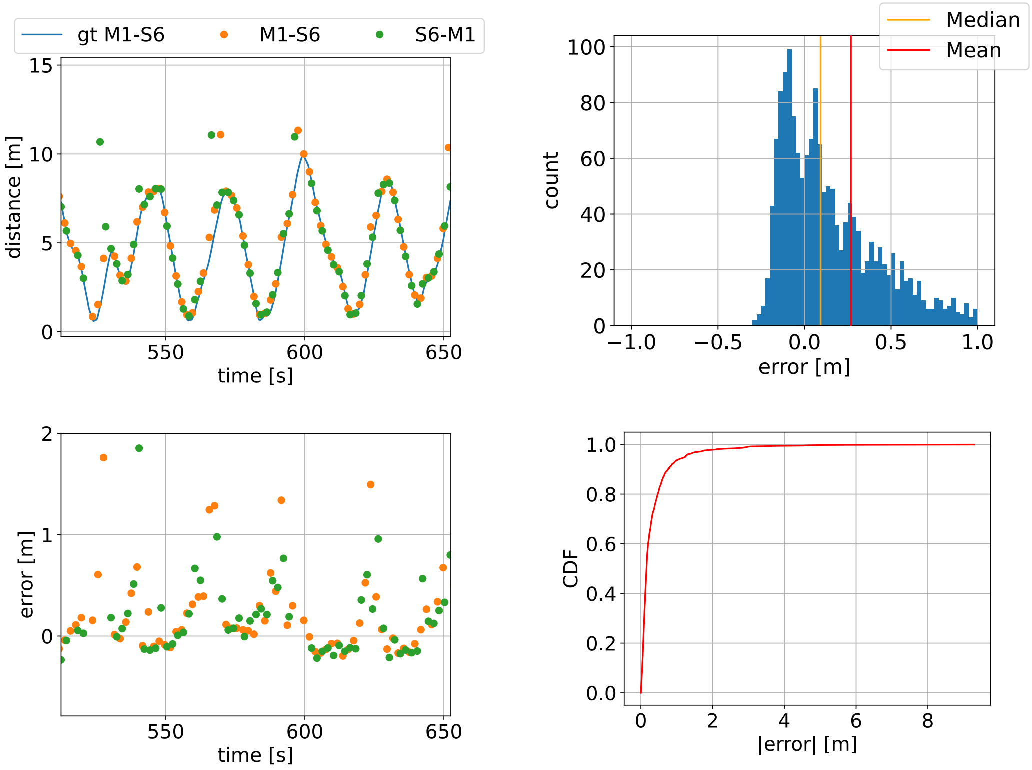
(b) Histogram and CDF of errors for all Line experiments.

Fig. 10. Dissecting proximity: Line.

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, an effect that has not been well studied in the literature. 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 analyze the effect at scale of these perturbations. We still employ only one stationary and one mobile tag, but this time the tester carrying the latter moves 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 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 11a) shows a pattern similar to the one in Line (Figure 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 often corresponding to changes in direction. Nevertheless, by comparing the histograms of



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

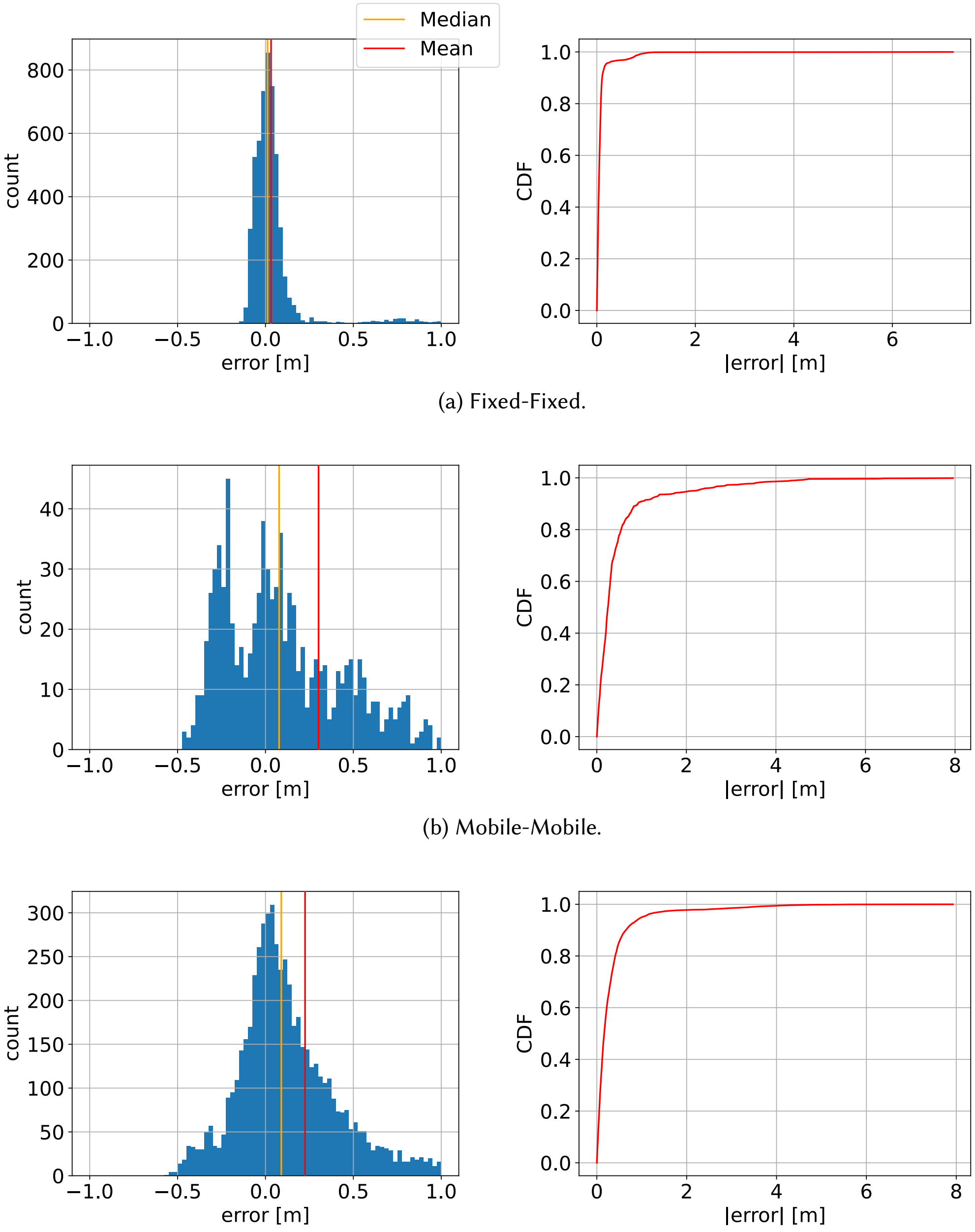
(b) Histogram and CDF of errors for for Random experiments.

Fig. 11. Arbitrary paths: Random.

this Random experiment (Figure 11b) vs. the Line one (Figure 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. Our final experiment involves 9 tags. It both enables us 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 9; these nodes may represent office workers sitting at desks or designated objects of interest. The remaining 3 tags are carried by testers; they start near the corners of the room, move over to cluster around node S1, and pause at ∼1.5 m from one another for 15 s, as if to have a conversation. Finally, they depart in an unconstrained way (Figure 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 12a) and observe that the results are well aligned with those in Figure 9, except for the appearance of a few, significantly overestimated distances. This is due to the body shielding randomly induced



(a) Fixed-Fixed.

(b) Mobile-Mobile.

(c) Mobile-Fixed.

Fig. 12. Many tags: Multiple.

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.

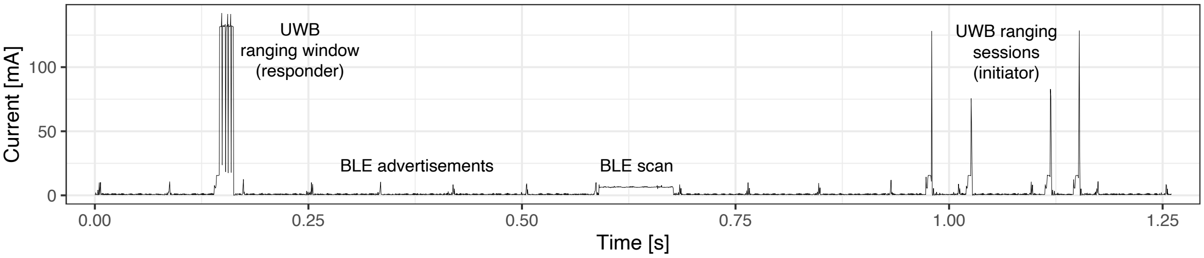


Fig. 13. Current draw for a Janus tag surrounded by 4 others.

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 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 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 one (Figure 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.

7.4 Energy Consumption

We investigate energy consumption, and therefore the lifetime achieved by Janus, by acquiring current draw measurements with a Keithley SourceMeter 2450. We use our custom tags (§6), as they do not suffer from the power limitations of the MDEK1001 devices.

This also provides us the opportunity to peek at the actual behavior of a Janus tag executing the protocol schedule (§3) for which we show a trace segment (Figure 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 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 explore 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 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

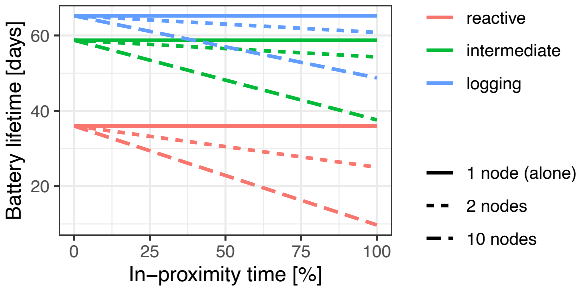


Fig. 14. Estimated battery lifetime for a Janus tag vs. the percentage of time spent in proximity as a function of the time ratio spent in communication range with one or nine other devices.

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 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.

These values can be further improved, as the current draw with both radios deactivated is relatively high, 0.48 mA. This can be reduced by fine-tuning the interaction with peripherals and other low-level aspects we did not address, focusing on optimizing the radio behavior. Still, even with this energy overhead, the lifetime we report for Janus is remarkably higher than existing research prototypes and market products (§10).

8 JANUS IN ACTION: EXPERIENCES WITH COVID-19 SOCIAL DISTANCING

We now complement our system evaluation (§7) by providing concrete examples of how Janus, and the excellent accuracy it offers, can be useful in a practical context. In comparison, here we cannot 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. On the other hand, the experiences we describe here are real-world ones; 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.

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 (§2) these social distancing scenarios combine both reactive and logging use cases, with the former specifically supported in our custom tag (§6) by audible alarms triggered when a safe distance is violated between two users. In all experiments described here, proper procedures were followed to recruit participants, compliant with GDPR and host organization regulations.

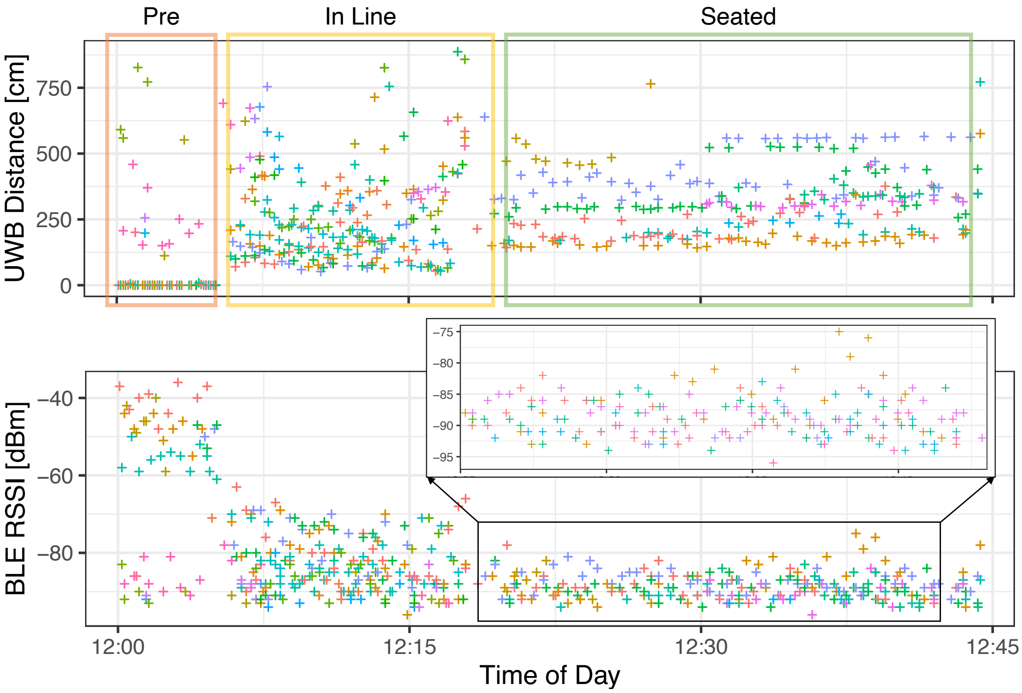


Fig. 15. Cafeteria: Raw data from one individual. The zoomed-in area shows detail of BLE data.

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

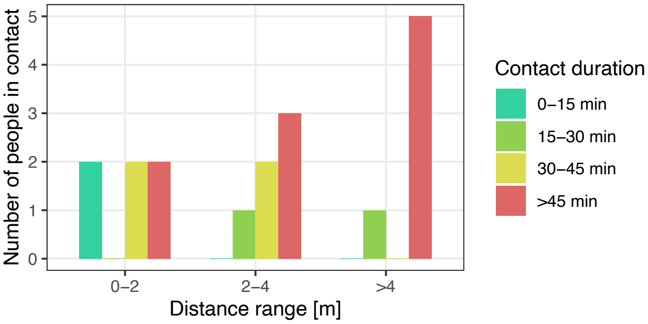


Fig. 16. Small office: Cumulative time of one individual at given distance from others during one workday.

8.2 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 our system evaluation (§7), by using the logging configuration with the same target parameters in Table 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 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*.

8.3 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 use the exact same intermediate configuration in Table 1. Further, we use a BLE TX power of 0 dBm, as this provides a good balance in this scenario with few people but several obstacles. Figure 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.

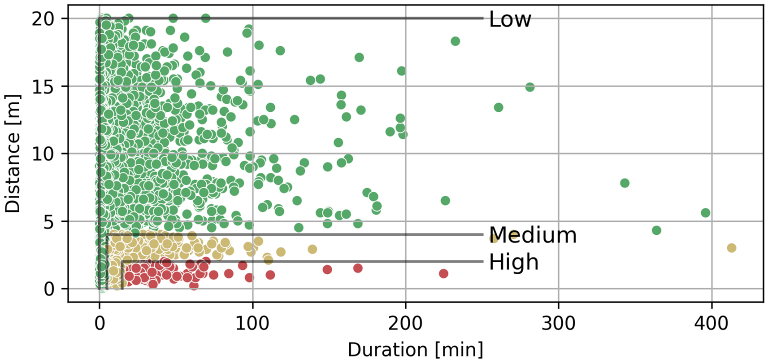


Fig. 17. Company-wide: Contacts of 90 individuals over 3 days.

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

To illustrate its power, enabled by Janus, we report 3 days of data collected with the intermediate configuration (Table 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 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.

8.5 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 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 (§6), to record only high-risk contacts and offload them

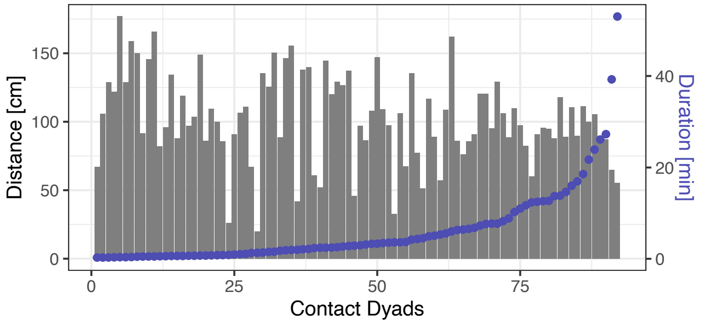


Fig. 18. Factory floor: average distance and total time in contact over a 24-hour period for each reported contact dyad.

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 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.

9 DISCUSSION

Our combination of system evaluation (§7) and real-world experiences (§8) allowed us to evaluate Janus with increasing degrees of complexity, covering all the requirements and design goals (§2) 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 (§4), validated through system evaluation, also shows an inherent tension between scale and latency. Our reactive configuration, with a discovery latency of $\Lambda = 2$ s, is currently in use 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 (§3) may help, scenarios with very low latency were not considered as part of those motivating Janus (§2) given that frequent ranging is inherently at odds with energy efficiency.

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 (§8). 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 [20]) 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.

10 RELATED WORK

Proximity detection has been an active field of research for a long time. The technologies exploited are numerous and include infrared [19], ultrasound [24], RFID [18, 34], and IEEE 802.15.4 [31] to name a few. Some comparisons focus on specific application fields [13], analyzing performance and deployment tradeoffs. In the following, we focus on Bluetooth and UWB, the technologies exploited in this paper. Moreover, in recent years Bluetooth has often been the technology of choice in many contexts, 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 [23], social studies [11, 32], museums [40], and proximity warning systems [13]. Many of these are implemented via custom solutions with emphasis on the application needs and little analysis of the fundamental properties or limitations of the radio technology selected. Further, the protocols developed are often simplistic and proximity measurements rely on the radio signal strength indicator (RSSI) reported upon reception of BLE advertisements to estimate distance via its relation to signal attenuation. This technique, however, is notoriously affected by environmental conditions and as a result yields significant estimation errors [42] that limit BLE to a narrow set of applications.

10.1 Contact Tracing Apps: Proximity Detection for the Masses

Paradoxically, the recent COVID-19 pandemic made the notion of BLE-based proximity detection commonplace even among the general public, due to a surge of contact tracing smartphone apps, whose use is often recommended, if not mandated, in many countries [12].

Many of these apps rely on Google-Apple Exposure Notification (GAEN) [4], an OS-level mechanism that, as a concerted effort by two major market players, offers a formidable penetration in the smartphone user base. Interestingly, its operation schedule is *fixed*: each phone emits a BLE advertisement every ~250 ms and scans for those from other phones every 4 minutes. The latter is a rather high value w.r.t. what is typically observed in the literature, even for logging applications, dictated by the need to provide a one-size-fits-all solution across the entire installed base and to tradeoff timeliness for energy-efficiency. Moreover, GAEN inherits the poor accuracy of BLE-based approaches, as recently assessed empirically in real environments [28]. These limitations notwithstanding, GAEN-based and other contact tracing apps have been instrumental in providing valuable data about the pandemic, and in bringing the opportunities of proximity detection to the forefront of society.

10.2 Proximity Tags: BLE or UWB?

Albeit pervasive, smartphones are not the only (or the best) devices enabling proximity detection. Not everyone owns them (e.g., many children and elders) and those who do may be reluctant to give access to them, e.g., due to privacy concerns. These considerations are at the core of another interesting trend amplified by the pandemic. Several commercial products already available as stand-alone proximity tags for both reactive and logging applications, have been re-targeted for contact tracing—and vice versa. Unlike smartphone applications, these specialized devices target situations where the use of tags can be controlled (or even enforced based on safety concerns), e.g., in a construction site, a factory, or other organizations.

The majority of these proximity tags is again based on BLE, from which they inherit the benefits and drawbacks outlined above, with poor accuracy notably among the latter. However, UWB tags are rapidly gaining momentum, thanks to the superior accuracy offered by this technology. This is largely popularized by full-fledged localization solutions that, nonetheless, require an infrastructure. Although some of these have been pitched as proximity detection solutions [5, 9], several others instead exploit an infrastructure-less architecture where proximity tags measure *directly* the distance between two devices via standard [10] or custom-designed *ranging* schemes. Unfortunately, the significantly higher energy consumption of UWB w.r.t. BLE significantly hampers the device lifetime. For instance, the Bump [2] system recently made the news as its use was required by all athletes and staff participating in the London marathon. However, it claims only 12 hours of operation [2]; many others fail to report lifetime altogether. Among these, is also the work in [17], again motivated by the recent pandemic, which highlights another challenge posed by UWB: the need to avoid collisions among ranging exchanges. The scheme proposed by the authors exploits a custom ranging protocol with probabilistic scheduling, which nonetheless is reported to successfully complete only 65% of the exchanges even with only 10 nodes.

Janus does not propose a new ranging scheme and is therefore virtually equivalent in accuracy to the UWB approaches above, which nonetheless rarely report their ranging techniques; in this respect, we use the most recent and performant scheme suggested by Decawave [22]. Nevertheless, in contrast, Janus achieves a much higher, and *predictable* success rate by virtue of its dual-radio approach, specifically by piggybacking on the out-of-band BLE channel information to *coordinate* the UWB ranging exchanges. Moreover, in contrast with BLE or UWB tags, Janus exploits the advantages of both, striking unprecedented tradeoffs between the key dimensions of accuracy and energy-efficiency.

10.3 Exploiting BLE and UWB in Synergy

This dual-radio approach is largely novel in both research and commercial devices. Among the latter, a few [6, 8] 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 [27] 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). 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 [16] 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 [16], unlike the analysis we provided (§7).

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 (§7.3). 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 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.

11 CONCLUSIONS AND OUTLOOK

We presented 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 ~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. Yet, the literature lacks accurate and realistic evaluations to the extent we provided here. We hope that these will inspire follow-up work targeted at mitigating these factors, e.g., by borrowing from the literature focusing on improving non-line-of-sight scenarios [14, 26] or by complementing UWB with the RSSI information already available from BLE [27].

These challenges and opportunities are likely to become even more relevant in the short term, given the rapidly increasing market penetration of UWB witnessed by the growing number of smartphones equipped with it alongside BLE. In this respect, we note that Janus does not rely on hardware-specific features of either radio chip, therefore porting it to the smartphone domain is conceptually simple, although ultimately determined by the specific APIs available. Ports of BLEnd as application space libraries already exist in the context of the Protect Texas Together project [7] for both Android and iOS; in the latter, protocol guarantees are met only when running in the foreground, due to limitations of the iOS BLE stack. As for UWB, smartphone APIs are still in the making. Nevertheless, they must provide a way to perform ranging against another node, which is the only essential UWB feature Janus relies on. Based on these considerations, we argue that the contribution described here is applicable, if not directly portable, to existing and upcoming UWB devices, extending the impact of our solution to the wider user base and use cases enabled by smartphones.

ACKNOWLEDGMENTS

This work is partially supported by Fondazione VRT, by EIT Digital (Project: ProxyAware, Activity: 20666) and by the Italian government (Project: NG-UWB, MIUR PRIN 2017).

REFERENCES

[1] [n.d.]. *Apple iBeacon.* Retrieved November 16, 2021 from https://developer.apple.com/ibeacon/

[2] [n.d.]. *Bump.* Retrieved November 16, 2021 from https://www.bump-space.com/

[3] [n.d.]. *Google Eddystone.* Retrieved November 16, 2021 from https://github.com/google/eddystone/

[4] [n.d.]. *Google/Apple Exposure Notification (GAEN).* Retrieved November 16, 2021 from https://covid19.apple.com/contacttracing/

[5] [n.d.]. *Pozyx.* Retrieved November 16, 2021 from https://www.pozyx.io/industries/health-and-safety/

[6] [n.d.]. *ProDongle.* Retrieved November 16, 2021 from https://www.prodongle.com/en/solution/social-distancing/

[7] [n.d.]. *Protect Texas Together.* Retrieved November 16, 2021 from https://protect.utexas.edu/app/

[8] [n.d.]. *Ubudu.* Retrieved November 16, 2021 from https://www.ubudu.com/social-distancing-assistant-for-enterprises-to-provide-a-safe-workplace-until-covid-19-goes-away/

[9] [n.d.]. *Wipelot.* Retrieved November 16, 2021 from http://socialdistancingtracing.com/

[10] 2011. *IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs).*

[11] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. 2011. Social fMRI: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing* 7 (2011).

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

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

[14] V. Barral, C. Escudero, J. García-Naya, and R. Maneiro-Catoira. 2019. NLOS Identification and Mitigation Using Low-Cost UWB Devices. *Sensors* 19, 16 (Aug. 2019).

[15] C. Beaulieu. 2004. Intercultural Study of Personal Space: A Case Study. *Journal of Applied Social Psychology* 34, 4 (2004).

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

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

[18] C. Cattuto, W. Van den Broeck, A. Barrat, V. Colizza, J. Pinton, and A. Vespignani. 2010. Dynamics of Person-to-Person Interactions from Distributed RFID Sensor Networks. *PLOS ONE* 5 (2010).

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

[20] M. Collotta, G. Pau, T. Talty, and O. Tonguz. 2018. Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Communications Magazine* 56, 7 (July 2018), 125–131.

[21] P. Corbalán and G.P. Picco. 2020. Ultra-wideband Concurrent Ranging. *ACM Trans. on Sensor Networks* 16, 4 (Oct. 2020), 1–41.

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

[23] S. Greenberg, N. Marquardt, T. Ballendat, R. Diaz-Marino, and M. Wang. 2011. Proxemic Interactions: The New Ubicomp? *ACM Interactions* 18, 1 (2011), 42–50.

[24] W. Huang, Y. Kuo, P. Pannuto, and P. Dutta. 2014. Opo: A Wearable Sensor for Capturing High-Fidelity Face-to-Face Interactions. In *Proc. of SenSys.*

[25] C. Julien, C. Liu, A.L. Murphy, and G.P. Picco. 2017. BLEnd: Practical Continuous Neighbor Discovery for Bluetooth Low Energy. In *Proc. of IPSN.*

[26] J. Khodjaev, Y. Park, and A. Saeed Malik. 2010. Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments. *Annals of telecommunications* 65, 5-6 (June 2010), 301–311.

[27] J. Kolakowski, V. Djaja-Josko, M. Kolakowski, and K. Broczek. 2020. UWB/BLE Tracking System for Elderly People Monitoring. *Sensors* 20 (2020).

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

[29] Decawave Ltd. 2016. DW1000 Device Driver Application Programming Interface (API) Guide.

[30] Decawave Ltd. 2017. DWM1001—Datasheet. Retrieved November 16, 2021 from https://www.decawave.com/dwm1001/datasheet/

[31] C. Martella, A. Miraglia, M. Cattani, and M. van Steen. 2016. Leveraging Proximity Sensing to Mine the Behavior of Museum Visitors. In *Proc. of PerCom*.

[32] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer. 2017. A Study of Bluetooth Low Energy performance for human proximity detection in the workplace. In *Proc of PerCom*.

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

[34] L. Ozella, F. Gesualdo, M. Tizzoni, C. Rizzo, E. Pandolfi, I. Campagna, A.E. Tozzi, and C. Cattuto. 2018. Close encounters between infants and household members measured through wearable proximity sensors. *PLOS ONE* 13 (2018).

[35] G.P. Picco, D. Molteni, A.L. Murphy, F. Ossi, F. Cagnacci, M. Corrá, and S. Nicoloso. 2015. Geo-referenced Proximity Detection of Wildlife with WildScope: Design and Characterization. In *Proc. of IPSN*.

[36] J. Staiano, B. Lepri, N. Aharony, F. Pianesi, N. Sebe, and A. Pentland. 2012. Friends Don't Lie: Inferring Personality Traits from Social Network Structure. In *Proc. of UbiComp*.

[37] D. Vecchia, P. Corbalán, T. Istomin, and G.P. Picco. 2019. Playing with Fire: Exploring Concurrent Transmissions in Ultra-wideband Radios. In *Proc. of SECON*.

[38] R. Want, B. N. Schilit, and S. Jenson. 2015. Enabling the Internet of Things. *IEEE Computer* 48, 1 (Jan. 2015), 28–35.

[39] C.C. Wilmers, B. Nickel, C.M. Bryce, J.A. Smith, R.E. Wheat, and V. Yovovich. 2015. The golden age of bio-logging: how animal-borne sensors are advancing the frontiers of ecology. *Ecology* 96 (2015).

[40] Y. Yoshimura, A. Krebs, and C. Ratti. 2017. Noninvasive Bluetooth Monitoring of Visitors' Length of Stay at the Louvre. *IEEE Pervasive Computing* 16, 2 (March 2017).

[41] F. Zafari, A. Gkelias, and K. K. Leung. 2019. A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials* 21, 3 (2019), 2568–2599.

[42] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy. 2016. Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons. *Sensors* 16 (2016).