IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2025, accepted 1 July 2025, date of publication 10 July 2025, date of current version 17 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3587489

RESEARCH ARTICLE

Self-Organizing Neighbor Discovery and Ranging for Battery-Powered Ultra-Wideband Anchors

DAVIDE VECCHIA[1], PABLO CORBALÁN[1], TIMOFEI ISTOMIN[1], GIAN PIETRO PICCO[1], (Senior Member, IEEE), AND ENRICO VARRIALE[2]

[1]Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy
[2]Thales Alenia Space Italia, 00131 Rome, Italy

Corresponding author: Gian Pietro Picco (gianpietro.picco@unitn.it)

This work was supported in part by Italian Government through the NG-UWB Project under Grant MIUR PRIN 2017; and in part by European Union—Next Generation EU through Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, partnership on "Telecommunications of the Future" (PE00000001—program "RESTART") under Grant CUP B53C22003970001.

ABSTRACT Ultra-wideband (UWB) technology has emerged as a popular solution in support of autonomous navigation. Unfortunately, installing a localization system is still a significant challenge in many large-scale or secluded environments. Since most available solutions require anchors to be connected to the power grid, infrastructures turn up to be extremely expensive when not simply unfeasible. We address these issues with SONAR, an UWB localization system based on battery-powered anchors. SONAR supports multi-year deployments without wired backbones, covering a wide spectrum of scenarios: from planetary exploration, our motivating application, to more earthly uses in, e.g., agricultural fields, smart warehouses, or smart health. The proposed system can be easily tuned to satisfy different needs in terms of ranging rate and energy consumption. We evaluate SONAR in a multi-hop testbed, showing that multiple roaming users can quickly discover each other and the surrounding anchors, and self-organize in a contention-free ranging schedule.

INDEX TERMS Autonomous vehicles, battery-powered anchors, energy efficiency, ultra-wideband.

I. INTRODUCTION

Localization is one of the key enablers of autonomous navigation. With GNSS services, a position fix can be obtained almost ubiquitously, and researchers have tackled the blind spots of satellite signals (e.g., indoor, in urban areas, under thick foliage) through dedicated infrastructures. Among dedicated solutions, ultra-wideband (UWB) has steadily gained popularity due to its decimeter-level accuracy in distance estimation (*ranging*) and its multipath rejection capabilities in harsh environments. Unfortunately, infrastructure deployment is a major challenge for localization systems, and those based on UWB are no exception. The installation of anchors is complex and expensive: their positions must be estimated accurately, their radios must be calibrated, and a wired backbone is typically necessary to power them and to communicate with remote servers. It is no coincidence that self-localization and self-calibration algorithms have received a lot of attention in recent years [1], [2], [3], tackling the hurdles of anchor setup—at least on the configuration side. Yet, wired connections are still a necessity in practice due to the high energy consumption of UWB chips. As a result, the use of UWB is precluded when installing mains-powered anchors is not an option.

To overcome this limitation, recent works made strides towards energy-efficient UWB anchors [5], [6], [7]. However, existing solutions either do not scale in terms of covered area, yield a short anchor lifetime, or impose high latencies (§II). Therefore, an open question remains about how to strike the best tradeoff among the energy consumption, update rate, and responsiveness of the anchors.

In this paper we present SONAR (Self-Organizing Neighbor discovery And Ranging), an energy-efficient UWB protocol for ranging-based localization using battery-powered

The associate editor coordinating the review of this manuscript and approving it for publication was Zhengmao Li.

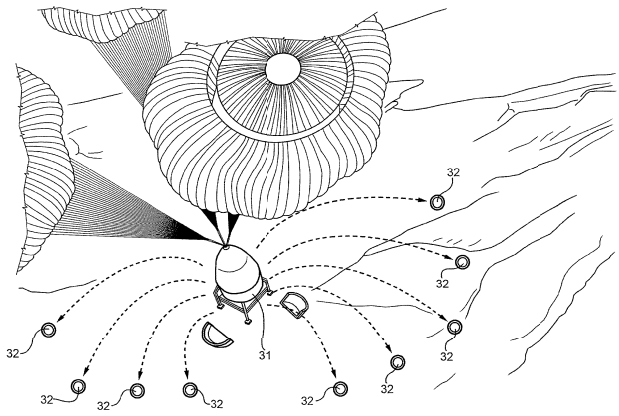


FIGURE 1. A lander ejects the UWB anchors that will constitute the localization infrastructure. Figure from [4].

anchors. SONAR finds application in all scenarios that require a deployment that cannot be feasibly connected to a communication backbone or the power grid, either due to costs or the impracticability of the environment. Indeed, our motivating application targets the quintessential example of impracticable environment: the surface of Mars. SONAR is part of a joint effort with Thales Alenia Space Italia to enable localization in unknown or difficult environments, notably including other planetary bodies. The overall approach has been described in a patent filed by the company [4], and implemented in a prototype replicating the various actors of the system [8], [9]. In the envisioned approach, the lander ejects UWB anchors before reaching the planet surface, as in Figure 1. After a self-localization phase to fix the anchor coordinates, anchors interact with the rover(s) used to explore the surface, performing ranging and multilateration. The question is then how to minimize anchor consumption in this second phase to ensure that the system enjoys a long operational lifetime. SONAR is our proposed answer to this question.

To reduce energy consumption, the uptime of anchors must necessarily be kept to a minimum. To this end, the main characteristic of SONAR is that anchors rely on a particularly energy-efficient approach to neighbor discovery, necessary to identify the users in range, and enter low-power mode (*deep sleep* radio state) when no user to be localized is around. This is crucial when covering large areas, since anchors may be isolated for most of the time. On the other hand, this approach requires an energy-efficient mechanism for anchors and users to discover the presence of each other and resume localization. One possible solution is to maintain network-wide synchronization, which facilitates communication between users and anchors. However, this translates to a constant energy drain even when users are far from anchors, in addition to the energy cost for neighbor discovery.

The main contribution of this paper is the design of a protocol that extends the lifetime of battery-powered anchors by relying on user orchestration and synchronizing connected users and anchors only when needed. This is done with *no communication overhead*, as a byproduct of the main operations of the system, i.e., neighbor discovery and ranging. SONAR builds on the assumption that users have a large battery capacity. Since the energy consumption of the radio is negligible compared to that of the engine, the user radio in SONAR is always active, listening for incoming transmissions from anchors that advertise their presence. As the name of the protocol suggests, users in SONAR can organize a shared contention-free schedule for transmissions, i.e., a dynamic assignment of time slots to specific users and anchors. This avoids collisions and needless competition for access to the radio medium, dynamically accommodating the needs of multiple users when they co-exist in the same localization area.

While developed in the context of planetary exploration, SONAR is meant to be used on Earth as well. A prominent use case is autonomous navigation for tractors and UAVs in agricultural fields, where cabling is impractical and GNSS positioning is affected by vegetation and buildings—even when considering multiple constellations [10]. Similar considerations hold in environments in continuous evolution, e.g., construction or mining sites, where SONAR enables accurate tracking of workers to enhance their safety, without deploying a permanent infrastructure. Another potential application is navigation in smart warehouses [6], where battery-powered anchors offer a much more affordable and easier to deploy compared to a fixed infrastructure. Moreover, in large areas that are visited only seldom, anchors would remain quiescent and with minimal energy expenditure for most of the time, alternating short energy-efficient neighbor discovery and deep sleep mode, therefore reconciling the conflicting needs of full coverage and long lifetime. Similarly, in an assisted living scenario, the SONAR localization infrastructure could engage in ranging only upon a significant event, e.g., a person has fallen. In general, SONAR provides an "on-demand" anchor infrastructure, which can be quickly deployed temporarily, or replace traditional long-term ones, and even be activated only when needed. Moreover, it supports a wide range of scenarios through a few simple parameters that tune discovery latency, ranging rate, and energy consumption.

We evaluate SONAR in Cloves [11], a multi-hop UWB testbed. Multiple nodes act as users meeting and departing, which allows us to study the responsiveness of the protocol in various conditions. Our analysis confirms that SONAR allows users to quickly discover nearby anchors and establish a shared ranging schedule, while anchors remain in power-saving mode most of the time.

The remainder of this paper is structured as follows. In the next section, we provide an overview of state-of-the-art localization systems with a focus on the energy consumption of their anchors. We describe next, in §III, the core properties of the proposed protocol, and in §IV the design principles that allow SONAR to overcome the

limitations of traditional localization systems with always-on anchors. After highlighting relevant implementation details in §V, we report in §VI the estimation of the energy consumption of anchors and their battery duration. In §VII we discuss the results of our evaluation, analyzing the performance of SONAR in terms of time to synchronize, ranging reliability and ranging accuracy. Finally, we share our conclusions and outlook in §VIII.

II. BACKGROUND AND RELATED WORK

We provide an overview of localization approaches, to determine which is most suited for our use cases, then focus on existing systems that target energy-efficient anchors.

A. UWB LOCALIZATION SYSTEMS

Localization systems can be divided based on the type of measurements they rely on, which require specific messaging schemes; therefore, the approach followed directly impacts the scalability and energy consumption of anchors.

1) TIME OF ARRIVAL (ToA)

UWB two-way ranging (TWR) [12] exploits ToA values to estimate the time-of-flight of a signal between two nodes, which can be converted to distance. In the single-sided version of TWR (SSTWR), only two packets are used. A user, e.g., a UWB tag, can perform TWR with multiple anchors to self-localize, a staple solution used in many systems [13]. Ranging-based localization does not require any additional communication among anchors, and is therefore suitable for our use cases. Yet, scalability in terms of the number of users is limited, since each single user interacts with each anchor individually.

Scalability could be achieved by exploiting ''virtual ranging'' [14]. In this approach, user only listens passively to beacons as a set of anchors perform double-sided TWR (DS-TWR) with a reference node of the infrastructure. This scales to countless users, but is not easily applicable to large infrastructures. In our use cases, supporting multi-hop networks and large-scale areas is a priority w.r.t. scalability in terms of users.

2) TIME DIFFERENCE OF ARRIVAL (TDoA)

Systems based on uplink TDoA (UL-TDoA) minimize the number of packets per localization, using a single one sent by the user to the anchors [15], [16]. While this can greatly reduce the energy consumption of the user, it is difficult to reconcile with energy-efficient schemes for the anchors. First, anchors must be kept tightly synchronized to ensure the accuracy of TDoA measurements. In the absence of wired clock distribution, impractical in our use cases, the synchronization requirement can be satisfied using additional UWB transmissions. However, this entails either periodic or per-localization communication overhead. Second, the information needed for the TDoA computation is collected by the anchors; in our use cases, the user should instead self-localize. This would require additional transmissions from the anchors to inform the user of the recorded information, essentially turning the UL-TDoA messaging scheme to ranging in terms of energy consumption from the anchor perspective.

The issue can be avoided by reversing the direction of the localization beacons, directly from the anchors to the tag, using downlink TDoA (DL-TDoA). Different flavors of DL-TDoA exist. Anchors either transmit packets in a staggered [17] or in a concurrent fashion [18] towards users, which are responsible for computing time differences. The main feature of DL-TDoA is support for countless users, as anchors do not have to interact with specific targets. However, all anchor transmissions must be scheduled with accurate timing. This means that in staggered DL-TDoA, anchor radios must remain on until their assigned time slot for transmission, consuming energy. This is not an issue in concurrent DL-TDoA; unfortunately, this variant can only tolerate a small distance difference between anchors before their concurrent signals disrupt each other. These limitations make DL-TDoA techniques suitable for a single, connected set of anchors in controlled settings, but cumbersome when scaling to large areas.

3) ANGLE OF ARRIVAL (AoA)

The basic principle of AoA-based approaches is to measure angles between devices and combine angle measurements w.r.t. multiple anchors to perform localization [19]. Alternatively, AoA can be combined with TWR to perform localization with a single anchor [20]. Since users only need to interact with one anchor, this could be a promising solution in conjunction with anchor selection to implement load balancing.

While these approaches can greatly reduce the number of exchanged packets per localization, AoA systems do not easily achieve high accuracy; even small angle estimation errors can lead to large errors when anchors are far from the user. Considering our requirements, we focus on a TWR solution to ensure anchors are low-complexity, provide high accuracy, avoid synchronization overhead, and scale more easily to large operational areas.

B. BATTERY-POWERED ANCHORS

The concept of battery-powered anchors in UWB infrastructures is not new, but their energy efficiency has received little attention when compared to other metrics like localization accuracy, latency, and scalability. Most literature that focuses on energy consumption only does so for the tag, assuming that the infrastructure is mains-powered. Interestingly, the available works that introduce battery-powered anchors all exploit ToA (TWR) rather than TDoA or AoA, in agreement with our review of the localization flavors in UWB systems.

In [5], the authors propose a portable UWB system where anchors can be battery-powered. However, their focus is on UAV flight control and anchor autocalibration to simplify installation without compromising localization accuracy, and

not specifically on energy efficiency. With a 129 mW consumption (34.9 mA at 3.7 V), the anchors cannot be used for long-lasting deployments.

In [6], the authors suggest that tag-equipped users, drones for inventory management, orchestrate nearby anchors using a dual-radio approach. The UWB radio is used for ToA-based localization, deferring all operations other than ranging to a low-power sub-GHz module. Crucially, the UWB radio of anchors is kept in ''sleep'' mode when no drone is around. Anchors discover approaching drones by listening on the sub-GHz channel, where drones broadcast their presence. While the sub-GHz radio consumes one order of magnitude less power than the UWB radio, it has a significant impact on anchor lifetime, with a constant current consumption of 3.4 mA even when no ranging is performed.

Instead of using a second radio, the approach in [7] adopts a dedicated MAC layer for UWB based on low-power listening. This allows users to organize a short TDMA schedule, sufficient for one localization. The process involves multiple phases. First, users enter a contention period to take hold of the schedule. Then, the ''winning'' user starts an anchor discovery phase. It probes anchors, which are in a duty cycled listening mode. After the discovery phase, nearby anchors send an acknowledgement, ensuring bidirectional discovery. Only then the exchange for ranging is performed. Unfortunately, anchors incur significant energy expenditure during this process, most likely because the anchor UWB radio is kept idle (rather than in deep sleep) for the entire duration between discovery and ranging. Indeed, the approach appears suitable only for very low ranging rates; otherwise, the battery of anchors is depleted rapidly.

While this approach is not directly applicable in our scenarios, we observe that anchor consumption has been greatly reduced by offloading scheduling responsibilities to the users. However, because users are also assumed to be energy-constrained, many interactions are required before ranging to establish a contention-free anchor-user connection. Differently from previous works, in our target applications we exploit the always-on radios of users to perform synchronization and slot negotiation; we do so by reusing neighbor discovery and ranging packets, thereby minimizing anchor operations to achieve unprecedented anchor efficiency.

III. GOALS AND ASSUMPTIONS

Battery-powered anchors provide great flexibility when deploying the infrastructure. This flexibility does not come for free, but with its own set of challenges for protocol designers to tackle in the quest to preserve energy.

In 2D ranging-based localization, users typically perform distance estimation with three or more anchors with known coordinates to compute their own position. This is generally easy to do when anchors are always active, listening on the radio channel waiting for user requests. However, when the energy supply is limited, anchors must stay in low-power mode for most of the time, becoming unavailable for users. Therefore, a dedicated protocol must be designed for anchors to keep their radio on only when there are users nearby that want to perform ranging.

Users perform ranging exchanges with the anchors in range, e.g., to perform proximity detection or full-blown localization. However, if the area is large, nodes (users and anchors) are *not* necessarily in range of each other.

We introduce a system for the orchestration of neighbor discovery and ranging across multiple users interacting with battery-powered anchors. It is designed to ensure a long deployment lifespan, stretching to years of operation. The protocol is easily configurable to balance ranging rate, responsiveness of neighbor discovery, and energy consumption, while adapting dynamically based on the user needs.

SONAR achieves the aforementioned goals by implementing the following properties.

A. AGGRESSIVE DUTY CYCLING OF ANCHORS

We first observe that anchors should limit energy consumption, entering sleep mode whenever possible. Two opportunities arise. On one hand, depending on the size of the operational area and the number of roaming users, it is likely for an anchor to remain isolated (i.e., not in range of any user) for most of the time; its operation can therefore be aggressively duty-cycled. On the other hand, even when one or more users are around, the anchor radio is put to sleep whenever the user is not interacting with it, e.g., while engaged with other anchors.

B. CONFIGURABLE NEIGHBOR DISCOVERY AND RANGING

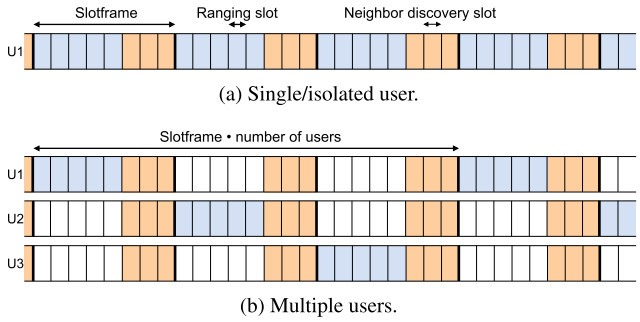
As users are moving, it is important that they rapidly discover the set of anchors that can be exploited for ranging. The discovery period depends on application and system requirements. Aiming for the lowest-latency discovery possible is unneeded in situations where users are slowly-moving, and therefore neighborhoods are mostly static. Likewise, different applications may require different ranging rates. By setting these two main parameters, the system can be easily adapted to different types of users, with predictable trade-offs w.r.t. energy consumption.

C. DYNAMIC TIME ALLOCATION FOR MULTIPLE USERS

When different users are operating in far-away, non-overlapping areas, each user can enjoy the entire communication bandwidth available for continuous, maximum-rate ranging. However, this is no longer possible when other users share the same medium. The system adapts to the contingent situation, e.g., dynamically throttling down the ranging rate of a user to make space for others or, on the contrary, throttling it up when they disappear and the user is alone again.

D. USER-SIDE ORCHESTRATION

One of the key aspects of SONAR is that the users, not the anchors, are responsible for ranging scheduling and



(a) Single/isolated user.

(b) Multiple users.

FIGURE 2. Slotframe ownership for an isolated user and for 3 co-located users. Neighbor discovery slots are shared.

contention resolution. To make this possible, we assume the radio of users is always on. Indeed, the energy drain of the radio is orders of magnitude smaller than the one of, e.g., the engines required to enable the user movement. Controlling the schedule on the user side also simplifies the logic for anchor selection. While outside the scope of this paper, it is easy to imagine that users may choose not to use anchors even when in range of them, limiting interactions for when they are really needed. Users may respond to discovery, and schedule ranging, only upon specific events, e.g., only when the user tag detects a fall, as in the assisted living scenario mentioned earlier (§I). Moreover, by including the battery level in anchor advertisements, a load balancing logic to extend the anchors lifetime can be applied unilaterally at the user, with no additional communication.

E. NO ASSUMPTION ON ANCHOR-TO-ANCHOR CONNECTIVITY

In the scenario of planetary exploration, the lander ejecting the anchors has virtually no control over their final placement, which means that the network topology cannot be known in advance. In agricultural fields and industrial settings, obstacles are likely to temporarily impair or block communication. Considering the challenging environments we target, we need to avoid assumptions on the quality of radio links between anchors; this is achieved by user-side orchestration, which removes at the core the need for anchors to ever communicate with each other.

IV. SONAR DESIGN

We first provide a general overview of the protocol, followed by the description of a single-user system, and finally introduce the changes required for multi-user mode.

A. HIGH-LEVEL PROTOCOL OVERVIEW

SONAR is based on a TDMA structure: the user coordinates ranging interactions with its neighboring anchors in a time-slotted fashion. However, the protocol does not rely on a global time reference or a global schedule. Slot scheduling and synchronization are performed *locally*, in the neighborhood of the user(s); at every iteration, each user can dynamically select its preferred group of anchors to range with.

The protocol comprises three main techniques for low-power operation. First, the anchors that are far away from any user keep their radio off most of the time, probing the medium periodically with a beacon to discover users. Second, the anchors that are serving one or more users follow the TDMA schedule which defines active and passive ranging slots for each anchor, thus letting them keep the radio off during time slots in which they are not involved. Finally, coordination among users has virtually no extra energy cost for the anchors. Users can self-organize, adjusting the ownership of time slots by exploiting ranging and neighbor discovery packets, without the need for any additional radio operation involving anchors.

Unlike anchors, users keep their radios always on to detect neighbor discovery beacons and to quickly coordinate the schedules among themselves once they overhear the activity of each other. If a single user is operating in an area, it can utilize all available time slots. Time slots are grouped into *slotframes* that repeat over time (Figure 2a). When different users approach, they switch to multi-user mode, where slotframes controlled by each of them alternate without overlapping (Figure 2b).

B. SINGLE-USER MODE

Single-user mode supports ranging and discovery involving any number of anchors, and provides the basis for the multi-user SONAR variant.

1) TIME-SLOTTED STRUCTURE

Time is split into *slots* of the same duration T_{slot}, long enough to fit the ranging handshake used in the system, i.e., single-sided two-way ranging (SSTWR), which requires two transmissions. Similarly, bidirectional neighbor discovery exploits a single slot hosting the anchor beacon, the user reply, and the anchor confirmation. Several consecutive time slots are grouped in a slotframe.

The number of slots per slotframe is a system-wide constant S, giving a slotframe duration $T_{sf} = ST_{slot}$. A configurable number of slots D at the end of the slotframe is reserved for neighbor discovery, leaving the first slots free to be used for up to $S - D$ ranging handshakes.

2) ANCHOR-INITIATED NEIGHBOR DISCOVERY

Initially, the anchors and the roaming user are not aware of each other. Anchors that are out of reach of any users (hereafter, *isolated anchors*) keep their radio off most of the time. To advertise their presence to the user, they periodically transmit a *discovery* beacon, `ND-INIT`, in a randomly selected slot not used for ranging. The interval between neighbor discovery attempts, T_{ND}, is an important parameter of SONAR anchors, balancing energy consumption and discovery responsiveness. After transmitting `ND-INIT`, the anchor waits for the short time needed for the user to reply with a `ND-RESP` packet. This includes the SONAR header

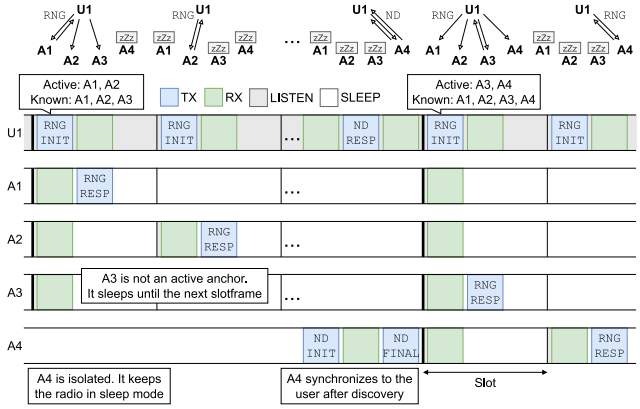


FIGURE 3. A slotframe schedule example, with one user, three known anchors and one to be discovered.

with synchronization information, described next; discovery is then confirmed to the user with the ND−FINAL packet.

3) USER-ANCHOR SYNCHRONIZATION

Anchors synchronize their slotframes to the one of the discovered user, adjusting their TDMA structure based on the content of ND−RESP. Synchronization information is expressed by the combination of slot number $s \in \{0, \ldots, S-1\}$ and time offset t_{offset} from slot start. Whenever the user sends a packet, it embeds t_{offset} as the difference between transmission time and slot start time. When the anchor receives the ND−RESP from the user, it timestamps the arrival of the packet at t_{RX}, exploiting the high-frequency clock of the UWB radio chip. This timestamp is used by the anchor for synchronization, simply performed as $t_{slot} = t_{RX} - t_{offset}$, yielding the slot start in UWB anchor time. The anchor can also compute the start of the next slotframe as $t_{sf} = t_{slot} + (S - s)T_{slot}$, exploiting the current slot number s.

Synchronization information also includes the current slotframe number f, necessary in the multi-user scenario (§IV-C) to assign slotframe ownership to the right user.

After discovery, anchors would naturally lose synchronization over time due to clock drift. To prevent it, anchors exploit ranging packets to (re-)synchronize and remain aligned to the user.

4) RANGING WITH DISCOVERED ANCHORS

While neighbor discovery is initiated by anchors, ranging handshakes are always initiated by the user. If the user is aware of at least one nearby anchor, it sends ranging requests starting in the first slot of the slotframe. Onto the ranging requests, RNG−INIT, users piggyback the same SONAR header as in ND−RESP packets, together with the scheduling information for the current slotframe. To receive the schedule, all time-synchronized anchors always wake up in the first slot of the slotframe and switch on the radio. When receiving a RNG−INIT packet, the surrounding anchors learn:

1) the synchronization information in the header, to align the slot structure to the advertised slotframe;
2) the list of *known anchors* that the user discovered;
3) the list of *active anchors* to be used for ranging in the current slotframe (a subset of known anchors);
4) the slot assignment for the current slotframe, or *ranging schedule*, i.e., which anchor is expected to reply in which slot to perform the ranging exchange.

The first anchor in the active list replies directly to the packet carrying the schedule; the others enter sleep to save energy after setting a wake up in the appropriate slot, which depends on their position in the list. The ranging exchange terminates when the active anchor transmits the RNG−RESP packet back to the user. If an anchor is not active but is known, it is a *passive anchor* in the current slotframe. Passive anchors sleep until the beginning of the next slotframe to receive the new schedule. In the example in Figure 3, the user U1 initiates a SSTWR handshake with anchor A1 and simultaneously announces that A2 is going to be requested next in the current slotframe while A3 is known to the user but will not be requested. In this case, anchors A1 and A2 are active, while A3 is passive.

Interestingly, the described scheduling approach allows for an arbitrary interleaving of the interactions with anchors. For instance, in Figure 3 the user could decide to perform, in the same slotframe, several consecutive ranges with A1 (e.g., to accumulate enough samples back-to-back) followed by multiple consecutive ranges with A2, or instead alternate between A1 and A2. Notably, users can select other users as ranging targets, instead of anchors. However, when multiple users can be present at once, their actions have to be first arranged so that they do not interfere with each other.

Besides active anchor selection by the user, Figure 3 also concisely illustrates user-anchor neighbor discovery and synchronization, covering the main operations in single-user mode.

C. MULTI-USER MODE

The approach described in the previous section is sufficient for the operations of a single SONAR user with any number of anchors. To support multiple users, the protocol must be extended to include inter-user discovery and synchronization. Figure 4 shows the high-level overview of the SONAR logic for users; the one for anchors is shown in Figure 5. We explore each component in detail next.

1) A LOOK AT SLOTFRAMES IN MULTI-USER MODE

Whenever a user discovers the presence of others, it shifts from single-user to multi-user mode. Instead of using all available slotframes, each user can range in a subset of them, interleaving operations with the others (Figure 2b). Slotframe ownership can be established locally by the user by exploiting its own ID and the current slotframe number f. For example, if four users are involved with IDs from 0 to 3, and the current slotframe number is 17, the owner is given by (17 mod 4), i.e., user 1.

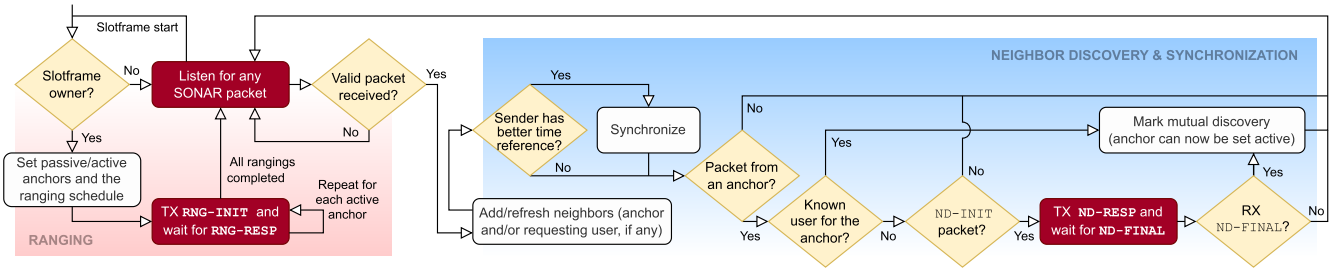


FIGURE 4. SONAR logic for the user. TX/RX operations are highlighted in red.

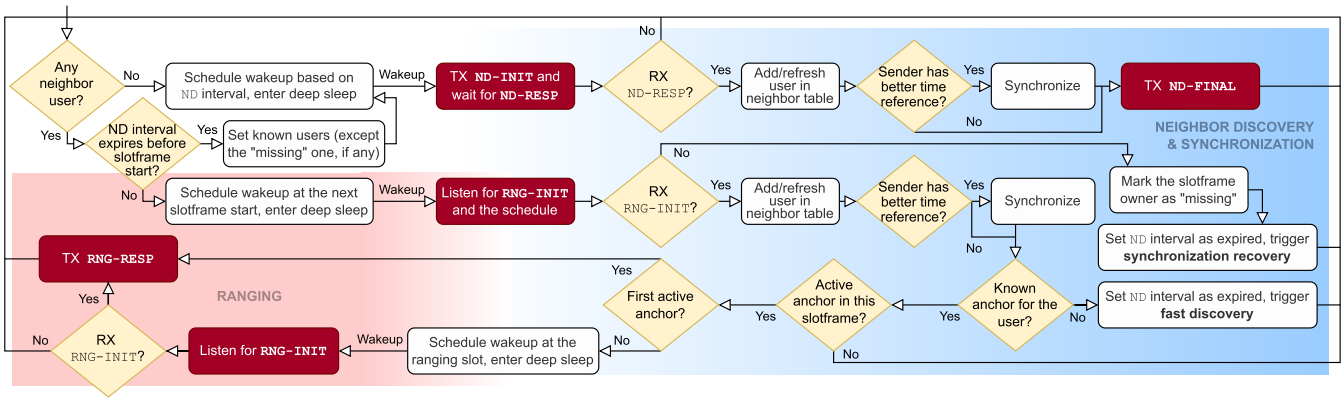


FIGURE 5. SONAR logic for the anchor. TX/RX operations are highlighted in red.

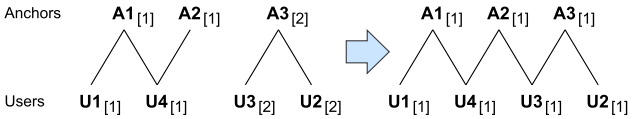


FIGURE 6. Time propagation when two groups merge. A nodes are anchors, U nodes are users. The value in brackets is the ID of the reference user. Multi-hop synchronization is performed through an anchor: A2 transmits a packet overheard by user U3, and the two groups align to U1.

For slotframe division to be effective, users must all agree on the current slotframe number and start time. However, because there is no global synchronization in the system, this requires dedicated mechanisms.

2) MULTI-USER NEIGHBOR DISCOVERY

To synchronize and avoid collisions, SONAR users must first be able to discover each other. The first step is to extend neighbor discovery, previously only performed between a single user and the anchors.

To ensure that an anchor discovers all surrounding users, the `ND-INIT` packet it broadcasts includes which users are already known to the anchor. This keeps known users from replying, allowing the missing ones to be discovered. Indeed, all users not yet discovered immediately send back a `ND-RESP` packet. While multiple overlapping replies could in principle cause a collision, one of them is likely to be decoded correctly by the anchor due to the properties of UWB concurrent transmissions [21]. Therefore, the anchor eventually discovers all users in range, possibly across multiple slotframes.

Next, we must also enable users to discover their peers. To avoid interference and needless contention, they must synchronize with all users interacting with the same anchors. Otherwise, two or more users might disrupt each other's attempt to range with a given anchor, with one repeatedly prevailing on the other based on their relative signal strength at the receiver. Still, even when the strongest packet is correctly decoded, ranging accuracy is at stake. Indeed, distance estimation is based on the time of flight (ToF) computed from TX and RX timestamps. The latter are obtained by the radio by inspecting the Channel Impulse Response (CIR), and identifying the *first path* in the signal. The overlapping of UWB preambles can cause a false first path to appear in the CIR, leading to incorrect distance measurements. Therefore, to prevent CIR interference, the relevant neighborhood of users is not limited to other users in direct communication range with them; rather, discovery is multi-hop, and must include *all* users in range with the same anchors.

This can be accomplished by exploiting the same `ND` and `RNG` packets already used. Relying on their always-on radios, users can learn about the presence of others either by:

- overhearing user packets (`ND-RESP`, `RNG-INIT`),

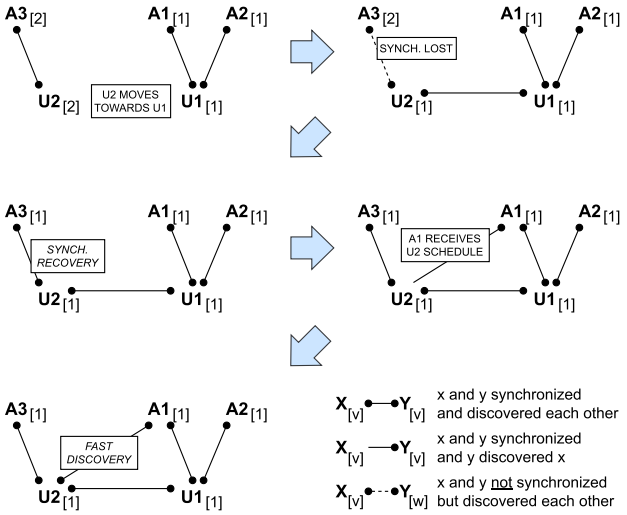


FIGURE 7. Merging of two groups of nodes. User U2 enters in the range of U1 and anchor A1. U2 synchronizes to U1, causing A3 to be left out-of-synch.

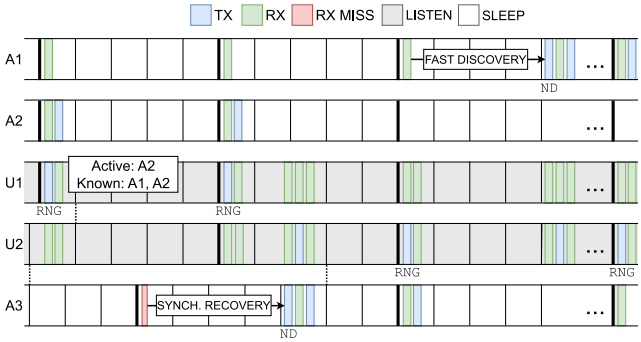


FIGURE 8. Slotframe view of synchronization recovery and fast discovery.A3 detects synchronization loss when no schedule is received at the beginning of the slotframe, which triggers synchronization recovery. Anchor A1 receives the schedule from user U2, but is not in the list of known anchors, which triggers fast discovery.

- receiving a neighbor discovery packet from an anchor (`ND-INIT`, `ND-FINAL`) or overhearing its ranging replies to other users (`RNG-RESP`).

Once users discovered their peers, they must synchronize before they can arrange a shared schedule.

3) INTER-USER SYNCHRONIZATION

When different users contend for the same anchors, they must eventually learn about each other. After discovery they must also agree on the source of synchronization, otherwise some user would end up unaligned and incapable of ranging with anchors. Ideally, the synchronization source is quickly re-established as users move, propagated by users and anchors alike, as shown in Figure 6.

As in user discovery, synchronization is performed in multi-hop by exploiting the same packets used for `ND` and `RNG`, and does not require additional transmissions. However, in multi-user the header with synchronization information, introduced in the single-user case, is included in *all* packets, not only those from the user (`RNG-INIT` and `ND-RESP`).

Indeed, while previously the only possible synchronization source was the only user, in this mode anchors become synchronization forwarders themselves, bridging the gap between nearby users.

As multiple synchronization sources are available, a mechanism is also needed to disambiguate them and determine which user should prevail, becoming the primary source all other nodes must align to. To this end, synchronization information is extended with the unique user ID indicating which user the sender of the packet has already aligned its slotframes to. A synchronization structure is created, starting from the primary source, based on the natural order in IDs. Users and anchors exploit this new information to align to the rest of their neighborhood, either accepting synchronization, if the source ID is lower than their own current source, or rejecting it otherwise. The user with the lowest ID becomes the *reference* the multi-hop neighborhood aligns to. In the example shown in Figure 6, the user group containing users U2 and U3 is initially bound to user U2 as the time reference. The group approaches and merges with another one, comprising U1 and U4. After the merge, the whole group binds to U1 as the new reference, and operations can continue without disruption. Finally, the time reference must also be accompanied by a freshness timestamp, to let receivers discard old information and avoid loops in time propagation.

These mechanisms ensure that a shared schedule is eventually established. However, discovery and re-synchronization are not instantaneous, and the system inevitably incurs in ranging interruptions when new users approach a group. We describe next two techniques used in SONAR to quickly restore time alignment and speed up discovery after a neighborhood change.

4) SYNCHRONIZATION RECOVERY AND FAST DISCOVERY

Ranging schedules may experience a transient inconsistency due to re-synchronization upon the arrival of a new user that has a lower ID. All anchors discovering the newcomer switch to the new reference, becoming unable to receive ranging requests from any other user. Equivalently, other users may synchronize to the newcomer and become out-of-synch w.r.t. anchors. The last case is depicted in the first and second step of Figure 7. User U2 comes close to U1. They discover each other and U2 synchronizes to U1 as a result. This causes anchor A3 to be left out-of-synch.

Instead of waiting for re-discovery, we tackle this issue with the *synchronization recovery* technique. When an anchor misses the `RNG-INIT` at the start of the slotframes, it detects a potential synchronization loss. Since it knows which user the slotframe belongs to, based on the current slotframe number f, the anchor marks that specific owner as "missing". The anchor can then recover synchronization in the same

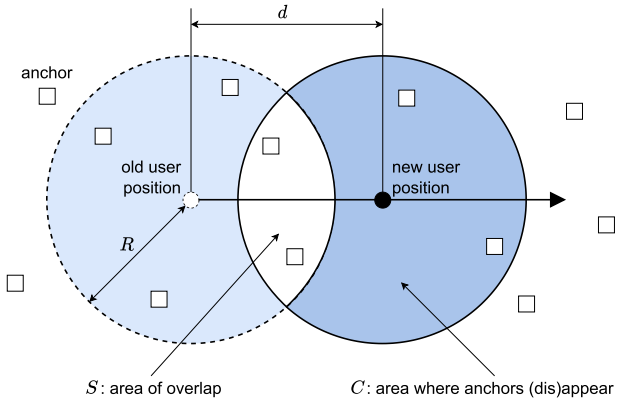


FIGURE 9. Model for neighbor churn.

slotframe by issuing a ND−INIT packet. However, it first removes the missing user from the list of known users embedded in the packet to trigger its reply. If the user has a new reference (like U2 in our example), the anchor exploits the synchronization information in ND−RESP to align to the new reference. Therefore, after only a *single* slotframe lost due to the transition, the anchor resumes normal operations.

With multiple users meeting and departing, we also see opportunities for a *fast discovery* procedure between users and anchors. We observe that anchors may be already synchronized to the user with the lowest ID before being discovered by all the others. This is often the case when a new user approaches anchors that are already synchronized to a user with lower ID, like in the last two steps of Figure 7. All nodes are already time-aligned to the same reference, U1, but U2 has yet to discover A1. The anchor (A1 in our example) can detect this circumstance by inspecting the list of known anchors when they receive the schedule embedded in RNG−INIT at the start of the slotframe (sent by U2). If the anchor cannot find itself in the list, it issues a ND−INIT in an appropriate slot of the current slotframe, even if T_{ND} has not expired yet.

The two techniques can be appreciated in the slotframe view of Figure 8. When U1 and U2 groups merge, A3 and A1 immediately become available for ranging with U2 by means of synchronization recovery and fast discovery, respectively.

D. ON THE RESPONSIVENESS OF NEIGHBOR DISCOVERY

There is clearly a tradeoff between discovery responsiveness and energy consumption, governed by the period T_{ND} with which the anchor transmits ND−INIT packets. The proper configuration of this parameter is application-dependent; we offer some intuition about it based on the expected number of new anchors encountered by a user. How fast a new anchor should be discovered is a function of the velocity of the user and the density of anchors. We present here a simple model that computes the number of neighbors changed (appearing/disappearing) per second based on the simplifying assumptions of *(i)* uniform distribution of anchors, *(ii)* constant velocity of the user, *(iii)* perfectly isotropic communication (unit disk graph model, UDG).

Figure 9 illustrates the main concept. Assuming a minimal distance l between anchors, the uniform density of anchors (anchors/m^2) is $m = 1/l^2$. The number of anchors in range of a given user is $n = mU$, where $U = \pi R^2$ is the space area around the user, assuming a communication range R. The area in which the number of nodes does *not* change is the intersection of the two circles in Figure 9, which can be computed as

$$S = 2R^2\cos^{-1}(\frac{d}{2R}) - \frac{1}{2}d\sqrt{4R^2 - d^2}$$

where d is the distance traveled by the user in 1 s.

The two areas in which nodes change are therefore $C = U - S$ each, and there are $c = mC$ nodes appearing (and c disappearing) or, dually, there is a neighbor change every $\frac{1}{c}$ seconds.

To put these numbers in context, with $R = 100$ m (typical for UWB outdoors), $l = 30$ m, and a velocity between 0.05 m/s and 15 m/s, we obtain the values in Table 1. The values 1.5 m/s, and 5 m/s are representative of human movement for walking and running, respectively, while the highest value of 15 m/s is similar to the maximum speed (54 km/h) of large vehicles used in agriculture, like tractors. The lowest value is included to represent the moving speed of a rover like Perseverance [22] showing that, in the planetary exploration case, neighbor discovery can be performed on a time scale of minutes, which translates to almost no power consumption while anchors are isolated.

TABLE 1. Sample values for the rate of neighbor change, to inform the periodicity of neighbor discovery.

Velocity (m/s)	c (#nbr changed)	$1/c$ (seconds between changes)
0.05	0.01	90
1.5	0.33	3.0
5	1.11	0.9
10	2.22	0.45
15	3.33	0.30

V. IMPLEMENTATION

We offer a detailed explanation of some prominent aspects of our prototype, that are however not part of the core concepts that constitute SONAR. This includes the implementation of active and passive anchor lists in the SONAR header, neighbor table management, and alternative ways to assign slotframes to users. We also provide insights on synchronization operations to motivate our choice for RX guard times, which affect energy consumption. SONAR is built atop Contiki [23] for the DecaWave EVB1000 platform [24] that hosts the DW1000 radio [25].

A. ACTIVE/KNOWN ANCHOR BITMAPS

When users broadcast the schedule they need to include a list with the active anchors for ranging within the slotframe and another list with the known anchors. To reduce packet length and transmission time, we implement these lists as bitmaps based on the node ID of each anchor. Hence, in a 32-anchor deployment, only 4 B are needed per list. Using bitmaps limits flexibility in anchor selection; in our prototype, anchors are used for ranging once per slotframe, and the order depends on the position of the bit associated to the anchor, i.e., the anchor associated to the n-th set bit in the active bitmap will be used in slot n. Still, considering that short IEEE 802.15.4 addresses are 2 B, using bit arrays can reduce significantly the packet length. In deployments with hundreds of anchors, or when the application requires full control of ranging requests, transmitting a list with short addresses or node IDs may be preferred.

B. SLOTFRAME DURATION AND ANCHOR RANGING RATE

Due to the bitmap-based schedule, an anchor will be used for ranging up to once per slotframe, as there is no way to encode multiple requests. Therefore, the slotframe duration T_{sf} also defines the maximum ranging rate of the anchor. For example, with $T_{sf} = 50$ ms, as in our evaluation, the maximum ranging rate is 20 Hz if the anchor is always active. This limit is not to be confused with the ranging and localization rate offered to the user. In our evaluation (§VII), we use $T_{sf} = 50$ ms, and 7 slots per slotframe dedicated to ranging, giving a maximum ranging rate of 140 Hz.

C. STRATEGIES FOR SLOTFRAME DIVISION

There are two main options to assign slotframe ownership to users. Based on our motivating applications, we opt for fixed assignment based on the total number of users in the system, best to ensure quick schedule reconciliation in closed systems. Another possibility is to assign slotframes based on the number of users in the same area, which enhances the ranging rate of co-located users and does not require knowledge of the set of users in advance. However, this option would increase the number of re-synchronization procedures, since the slotframe assignment would change upon the arrival or departure of *any* user.

D. NEIGHBOR TABLE

All nodes, users and anchors, host a neighbor table. Users store information for both discovered users and anchors, while anchors store only information related to users. Upon receiving a packet from a given node, its corresponding neighbor entry is created or refreshed. Each entry in the table includes the last-heard timestamp for neighbor expiration. Periodically, each entry in the table is checked to see if it has expired and therefore no longer valid, in which case the neighbor entry is removed. Expiration time should be carefully chosen to avoid the early removal of a valid neighbor. On the other hand, anchor entries in the user table must be refreshed, to allow users to perform ranging only with the anchors still in range. Similarly, anchors need to remove far users to return to isolated mode and save energy. Expiration time is application-dependent and is related to the chosen anchor selection policy. In our prototype, we set the equivalent of 3 T_{ND} for both user and anchor entries, to give nodes multiple chances to be heard before removal.

E. LOW-POWER SYNCHRONIZATION

If anchors could keep the UWB radio always on, no other mechanism except the one presented in §IV-B would be needed for synchronization. Unfortunately, precise synchronization is at odds with the need for SONAR to be as energy efficient as possible. To avoid the constant consumption of the DW1000 in idle mode, which can be as high as 18 mA, we set the radio to enter deep sleep mode ($<$100 nA) at the end of the slot. This means that we cannot exploit the 125 MHz digital PLL clock of the DW1000 (8 ns resolution) to maintain the time reference. To enforce the time-slotted operation, we resort to the 32 kHz MCU clock present on the EVB1000, which provides a much lower resolution (around 30 µs). Upon synchronization, after receiving a packet, we acquire the current PLL and MCU time to convert the RX timestamp to MCU time. The MCU clock can then be used to schedule the beginning and the end of each slot, firing an interrupt at a given future timestamp.

F. RX GUARD AND DURATION

Because of the limited resolution of the MCU clock, we set anchors awaiting `RNG-INIT` to start listening ~32 µs in advance. The DW1000 is then configured to search the radio medium for a preamble in the following 64 µs (preamble hunting phase). Due to the guard, this ensures that at least 32 µs are allocated to detect the preamble. If a preamble is detected in this very short period, the receiver remains active until the Start of Frame Delimiter (SFD) sequence is found, after at most 129 µs. If no preamble is detected, or if no SFD is found, the radio returns to deep sleep. The guard time applies only to the reception of `RNG-INIT`; when the anchor is waiting for `ND-RESP` after transmitting `ND-INIT`, the DW1000 has not yet entered deep sleep and can exploit PLL time. Therefore, no guard time is needed and preamble hunting is performed for 32 µs. These values for RX duration play an important role in the estimation of the energy consumption, in the next section.

VI. ENERGY CONSUMPTION

SONAR is designed to reduce the energy consumption of anchors when no user is around (*isolated*), but also while interacting with users (*active* and *passive* anchors). In this section, we break down the energy costs in these various modes of operation to provide an estimate of the lifetime that anchors can achieve. Since SONAR follows a simple slotframe structure, it is easy to compute the expected consumption by combining the cost of neighbor discovery, schedule reception, and ranging.

TABLE 2. Current draw and duration for each frame type. Write/read refer to the required radio operations in addition to TX/RX. Listen refers to the preamble hunting phase, and is only relevant for anchors, that receive `ND-RESP` and `RNG-INIT`. Idle duration is the portion of the response delay (before `ND-RESP` is sent, or waiting to transmit `RNG-RESP`) in which the anchor is not listening or receiving.

	ND-INIT	ND-RESP	RNG-INIT	RNG-RESP
Frame duration (µs)	196.86	190.71	199.94	205.06
TX average current (mA)	82.99	83.58	82.71	82.26
RX average current (mA)	131.80	131.68	131.86	131.96
Write duration (µs)	88	82	93	96
Read duration (µs)	75	69	81	84
Write current (mA)	15			
Read current (mA)	12			
Listen/miss duration (µs)	1 / 32	32 / 64		
Listen/miss current (mA)	118			
Idle duration (µs)	661			512
Idle current (mA)	18			
Wake up duration (µs)	5507			
Wake up current (mA)	3.01			
Deep sleep current (nA)	100			

A. DISCHARGE FOR NEIGHBOR DISCOVERY

The lost charge for one (successful) neighbor discovery slot is obtained as:

$$Q_{ND} = 2Q_{ND}^{TX} + 2Q_{ND}^{idle} + Q_{ND}^{listen} + Q_{ND}^{RX} \tag{1}$$

where the lost charge for each contribution is obtained by multiplying its duration by the associated current draw found in the DW1000 datasheet [25]. Energy expenditure is then obtained by multiplying the lost charge by the supply voltage of the radio, 3.3 V. Table 2 shows the duration of packets used in our evaluation, the average current draw per packet, and a breakdown of duration and current draw for various radio operations. The term $2Q_{ND}^{TX}$ represents the cost for the transmission of `ND-INIT` and `ND-FINAL`, including the writing of the frame into the radio buffer. Similarly, Q_{ND}^{RX} combines the cost of reception of the reply `ND-RESP` and the read operation from the radio buffer. Q_{ND}^{listen} is the discharge for preamble hunting, when the anchor is trying to detect an ongoing transmission; its value is close to zero, since we configure the radio to start listening exactly when `ND-RESP` is transmitted (§V). The term $2Q_{ND}^{idle}$ is the discharge when the radio is in idle mode, and depends on the delay between receptions and replies. In our prototype, this delay is set to 661 µs.

The calculation above yields the lost charge when a user replies with the `ND-RESP` packet. To estimate the consumption of isolated anchors (no user in the vicinity) we compute the cost of an *unsuccessful* discovery attempt as:

$$\hat{Q}_{ND} = Q_{ND}^{TX} + Q_{ND}^{idle} + \hat{Q}_{ND}^{listen} \tag{2}$$

TABLE 3. Energy consumption for an anchor in isolated, passive or active mode. Estimated duration of a 10.4 Ah, 3.7 V battery in each state and for mixed anchor activity patterns.

Slotframe duration T_{sf} (s)	0.05			0.5		
ND interval T_{ND} (s)	0.1	0.3	1	0.1	0.3	1
Energy consumption						
Isolated cost per slotframe (mJ)	0.084	0.030	0.010	0.845	0.296	0.104
Passive cost per slotframe (mJ)	0.242	0.187	0.168	1.002	0.453	0.261
Active cost per slotframe (mJ)	0.490	0.435	0.416	1.251	0.702	0.509
Lifetime						
100% isolated battery duration	2.4y	6.9y	19.7y	2.4y	6.9y	19.7y
100% passive battery duration	308d	1.1y	1.2y	2.0y	4.5y	7.8y
100% active battery duration	152d	171d	179d	1.6y	2.9y	4.0y
5% passive 5% active	1.8y	3.5y	5.3y	2.3y	6.3y	15.5y
20% passive 20% active	1.0y	1.4y	1.7y	2.1y	5.0y	9.4y
50% passive 50% active	204d	240d	255d	1.8y	3.5y	5.3y

where $\hat{Q}_{ND}^{listen}$ is a listening time that replaces $Q_{ND}^{listen} + Q_{ND}^{RX}$, due to the fixed duration of preamble hunting (32 µs) when reception does not occur.

B. DISCHARGE FOR SCHEDULE RECEPTION

Schedule reception is the only additional operation for passive anchors, which are not used for ranging. The schedule is effectively transmitted as part of the first ranging exchange of the slotframe. Therefore, the discharge combines a short listening, due to a guard time, and `RNG-INIT` reception:

$$Q_{SR} = Q_{RNG}^{listen} + Q_{RNG}^{RX} \tag{3}$$

C. DISCHARGE FOR RANGING

Finally, we estimate the cost of a ranging exchange, which comprises `RNG-INIT` reception and `RNG-RESP` transmission:

$$Q_{RNG} = Q_{RNG}^{listen} + Q_{RNG}^{RX} + Q_{RNG}^{idle} + Q_{RNG}^{TX} \tag{4}$$

Q_{RNG}^{idle} is the energy cost associated to the radio idle time before the `RNG` response, set to 512 µs, as it was found to be sufficient for all processing and SPI operations.

D. COMBINING ANCHOR STATES

We can combine the energy costs above to obtain the overall energy expenditure of an anchor. We assume that *(i)* an active anchor is such for all nearby users, in all slotframes, *(ii)* in each slotframe the anchor is used for ranging only once as in our implementation, and *(iii)* it is never the first anchor in the schedule. If the anchor is not always active, energy consumption decreases, approaching the estimate for the passive case. Similarly, consumption is reduced if the anchor is the first in the schedule and can immediately complete ranging in the first slot. While pessimistic, these assumptions simplify the overall estimation: energy consumption does not

depend on the number of users since the anchor is ranging once in all slotframes in any case.

Energy consumption depends on two parameters: the neighbor discovery interval T_{ND}, i.e., the wanted responsiveness of neighbor discovery, and the slotframe duration T_{sf}, which directly governs the ranging rate of the anchor under our assumptions. We also note that most of neighbor discovery attempts from the anchor will receive no response once all users have been discovered. Therefore, $\hat{Q}_{ND}$ is considered instead of Q_{ND}, and the one-time additional cost for successful discovery is considered to be negligible. Finally, the cost Q_{wu} to wake up the radio from deep sleep cannot be neglected; we add it to all the previous components. The resulting consumption over a given time interval D for each state can be computed as follows:

$$Q_{isolated} = \frac{D}{T_{ND}}(\hat{Q}_{ND} + Q_{wu}) \tag{5}$$

$$Q_{passive} = \frac{D}{T_{ND}}(\hat{Q}_{ND} + Q_{wu}) + \frac{D}{T_{sf}}(Q_{SR} + Q_{wu}) \tag{6}$$

$$Q_{active} = \frac{D}{T_{ND}}(\hat{Q}_{ND} + Q_{wu}) + \frac{D}{T_{sf}}(Q_{SR} + Q_{RNG} + 2Q_{wu}) \tag{7}$$

E. ANCHOR LIFETIME ESTIMATION

We quantify the expected lifetime of a SONAR anchor based on the discharge formulas for each state. The energy costs for individual operations are 0.165 mJ for $\hat{Q}_{ND}$, 0.157 mJ for Q_{SR}, and 0.248 mJ for Q_{RNG}. Given these costs, we compute the average energy cost per slotframe based on Eq. (5)–(7). The results for isolated, passive and active anchors are reported in Table 3.

In addition, Table 3 shows the expected lifetime when using a large battery providing 10.4 Ah at 3.7 V, compatible with our use cases. We do not account for the loss of battery capacity due to aging or environmental factors, e.g., temperature. In our analysis, we also consider the constant current consumption of the board that would host the DW1000. However, the EVB1000 platform used in our setup is an evaluation board whose consumption is not representative of the achievable performance. Therefore, we consider a 13 µA current consumption and a 93% power efficiency, following the same assumptions made by the manufacturer in a similar evaluation [26]. We considered two values for T_{sf} and three for T_{ND}, plus several activity patterns, from always isolated to always active anchors. The choice of T_{sf} depends on the desired ranging rate. T_{ND}, instead, controls neighbor discovery responsiveness (§IV-D).

With $T_{ND} = 0.3$ s, $T_{sf} = 50$ ms, and a low-activity pattern where the anchor is passive for 5% of the time and active for an additional 5% of the time, the battery is expected to last more than 3.5 years. This corresponds to an average ranging rate of 1 Hz. Even in a high-activity scenario, where the anchor is passive half of the time and active the other half, with an average ranging rate of 10 Hz, the battery is expected to last 240 days. Note that the reported ranging rate is averaged over the entire anchor lifetime to simplify comparison. However, while active, the anchor configured with $T_{sf} = 50$ ms performs ranging at 20 Hz, yielding a short time gap between the ranging exchanges used in localization, catering for the relatively fast-moving vehicles in our target scenarios.

When the mobile users are slower, SONAR can be adapted by changing T_{ND} and T_{sf}. With shorter neighbor discovery intervals and longer slotframes, the anchors would last many years. For example, with $T_{ND} = 1$ s and $T_{sf} = 500$ ms in a low-activity scenario (average ranging rate 0.1 Hz), the battery would last 15.5 years.

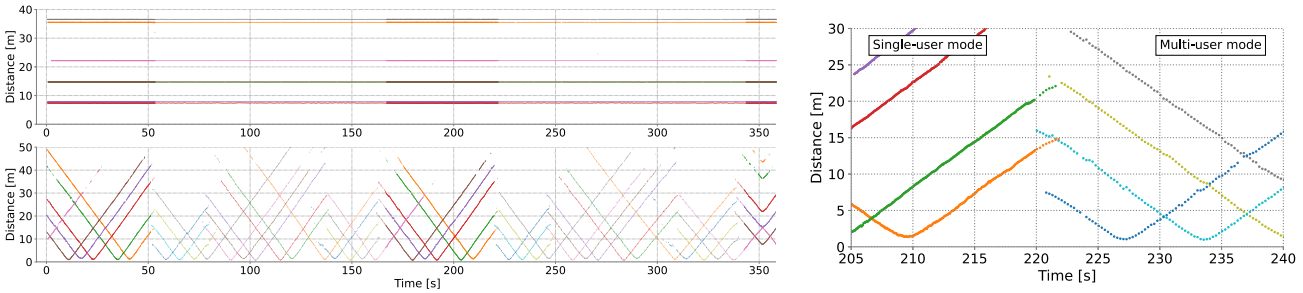
Finally, we look back at our motivating scenario, planetary exploration. For low-activity applications with very slow users (i.e., T_{ND} in the order of minutes, few ranging exchanges per minute), the average consumption becomes almost negligible. Therefore, given the small impact of SONAR on the battery, the UWB infrastructure could easily support additional services, like communication between the rover and the lander.

VII. EVALUATION

We evaluate SONAR in Cloves [11], a large-scale public UWB testbed, where we assess the ability of the system to quickly synchronize and dynamically define the ranging schedule.

We first show an execution example, with a person carrying an UWB tag representing the user (§VII-A). Then, we observe that ranging outliers may appear for neighboring nodes that have not discovered each other. To devise a solution, we analyze the relationship between communication range and interference range (§VII-B). Finally, we characterize quantitatively the robustness of neighbor discovery and slotframe alignment in benchmark experiments, with a selection of testbed nodes acting as anchors or users (§VII-C). Testbed nodes configured as users stop and resume operations in controlled time intervals, simulating arrival and departure. In multi-user benchmarks, we show that user synchronization is achieved quickly even in complex scenarios and that the ranging success rate in SONAR is essentially the same as the success rate of the baseline ranging application. These experiments explore the most adverse conditions that can affect SONAR, including the *simultaneous* arrival of up to five users in the same area.

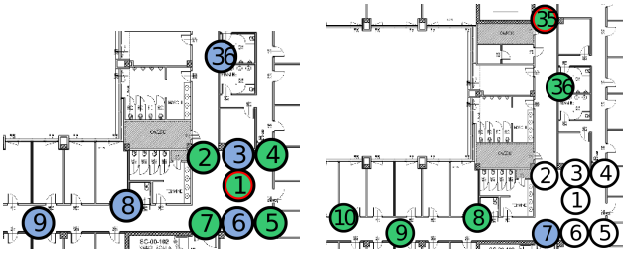
System Configuration: In all experiments, the DW1000 radio is set to use a 6.8 Mbps data rate, 64 MHz PRF, 128-symbol preamble. For the illustrative test in §VII-A we employ channel 2, whose long range better serves visualization. In the other experiments we switch to channel 5, whose higher central frequency yields shorter range and is more suited to explore various topologies in indoor settings. As for SONAR parameters, we set $S = 10$ slots per slotframe, and always leave the last $D = 3$ slots for neighbor discovery. Setting $T_{slot} = 5$ ms gives $T_{sf} = 50$ ms, which translates



(a) Ranging with anchors.

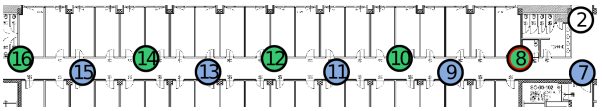
(b) Zoom-in of the mobile user.

FIGURE 10. A stationary user (above) and a mobile user (below) range with the neighboring anchors. When the mobile user discovers the same anchors used by the stationary one, they switch from single-user (ranging in all slotframes) to multi-user mode (ranging in a subset of the slotframes).



(a) *clique*.

(c) *disconnected*.



(b) *connected*.

FIGURE 11. Users (green) and anchors (blue) in the different connectivity scenarios. The reference user is highlighted in red.

to a maximum ranging rate of 140 Hz for users and 20 Hz for anchors.

A. MODE SWITCH IN MOBILITY TESTS

Users switch between single-user and multi-user mode depending on their respective location. To show this, a person carries a tag across the 36-node UWB testbed. This mobile user discovers and performs ranging with various anchors, as clearly shown by the V-shape of the estimated distance created by the user approaching an anchor then leaving (Figure 10a). The hand-held tag is one of two users in the experiments; the other user is one of the infrastructure nodes. We use only two users to clearly distinguish the zones where the users are interacting with different anchors, in single-user mode, from those where the two users meet, switching to multi-user mode. Given our SONAR configuration, the user can select up to 7 nearby anchors for ranging in the same slotframe, but each slotframe in multi-user mode is entirely assigned to a specific user. The transition between modes is more evident in Figure 10b, a zoom-in of the previous one, where the gaps in ranging on the right are due to slotframe sharing.

B. INTERFERENCE VS. NEIGHBOR DISCOVERY RANGE

In the experiments with mobile users we observed the appearance of outliers when users were departing. Because SONAR users and anchors are synchronized, the problem of CIR interference introduced in §IV-C might emerge if users previously co-located—and time-aligned—return to single-user mode. While they do not intentionally interact with the same anchors anymore, they can interfere with each other. Indeed, the interference range is larger than the communication range, therefore a synchronized node can disrupt the ranging of another one although it is not able to discover it. When this happens, unaware nodes in interference range transmit in the same slots, causing their signals to overlap. While overlapping UWB transmissions rarely cause payload corruption at the receiver (one of the payloads can typically be decoded correctly), the energy in the interfering preamble is still accumulated in the estimation of the CIR. This can cause an extraneous peak to appear in the CIR, compromising the first path detection algorithm and resulting in large ranging errors.

Several solutions can be considered. If users are always expected to be co-located, or increasing the ranging rate is not necessary, they could simply stay in multi-user mode and never switch back to single-user. Otherwise, the peculiar UWB feature of *complex channels*, defined on the same channel frequency but with different radio parameters [21], can be exploited to reduce interference by assigning a specific complex channel to each user. Finally, the system could perform neighbor discovery with greater transmission power, hence longer communication range, w.r.t. `RNG`. If the communication range of `ND` packets from anchors extends beyond the ranging interference range, then no user is affected by overlapping transmissions from distant anchors. Instead, any interfering anchor would be quickly discovered, which in turn would trigger multi-user mode again.

But what is the TX gain required by `ND` packets to achieve a communication range beyond the CIR-level interference range of `RNG` packets? We experimentally obtain an approximation by replicating the interference scenario with testbed nodes. The ranging initiator synchronizes two other nodes;

the recipient closer to the initiator is the ranging responder, while the other is the interference source. The responder is the farthest testbed node within RNG communication range w.r.t. the initiator. We observe a 13% success rate for ranging. For the interference source, we pick the farthest testbed node that negatively affects ranging results, as clearly noticeable from the standard deviation of measurements (>1 m). We perform 8000 ranging rounds and find that, with the responder at ∼29 m from the initiator, an interference source at ∼57 m or closer can negatively affect ranging. In SONAR, the interference source could be an anchor. Therefore, its ND packets should use a sufficient TX gain to reach the initiator of this experiment (i.e., a user). We found that a TX gain of 7 dB gives a reliable link towards the initiator (with a reception rate of 81%), and we use this value in our evaluation.

C. MULTI-USER MODE QUANTITATIVE ANALYSIS

When a new user approaches, other users and anchors in the vicinity may suffer temporary disruptions while they re-synchronize. Therefore, it is critical to evaluate the performance of SONAR during these transitions.

The example presented in §VII-A is not suited to extract quantitative results: we need a more controlled setup, with predetermined neighbor sets and known distances between nodes. We achieve this goal by exploiting the fixed nodes of the testbed to represent both users and anchors.

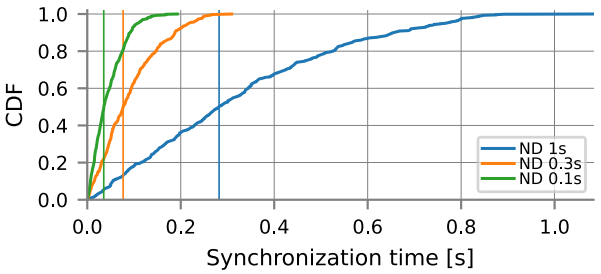
1) GOALS AND METHODOLOGY

We run many experiments in which several users appear almost simultaneously, contending for channel access in the attempt to range with the same group of anchors. The random arrival of users is "staged" in software. The experiment is divided into 200 rounds. At the beginning of a round, anchors and users are synchronized, but immediately apply a random delay between 0 and the duration of a slotframe (T_{sf}) to enforce a complete lack of synchronization. As users re-join the system in each round, they have to organize the schedule quickly to prevent collisions and spurious ranging results due to unwanted overlapping signals. Between rounds, users stop interacting with the surrounding nodes, making all of them "forget" their neighbors, thus artificially emulating users that leave the area to return later.

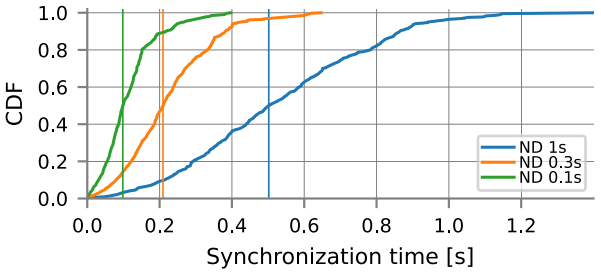
Our experiments produce the following indicators: 1) *synchronization time*, i.e., the delay between the appearance of a new user and its synchronization to the user with lowest ID, 2) *ranging success rate*, and 3) *distance estimation error*. For the ranging metrics, success rate and error, we compare against an application performing SSTWR. This baseline is acquired by scheduling each ranging exchange individually so that no other node is interfering.

Intuitively, synchronization time is affected by the ranging rate, as users can overhear the ranging exchanges of other nodes. However, the main factor is the neighbor discovery interval T_{ND}, even more so when users are not directly in range with each other. To explore the dependency of neighbor discovery on the network topology, we identify three relevant scenarios:

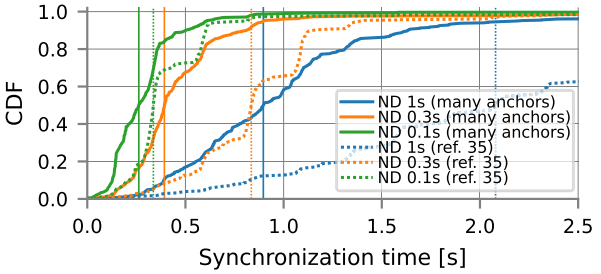
1) *clique*. The set of users forms a clique; as soon as a user begins ranging, all users can discover it by directly overhearing its ranging poll (RNG−INIT) or its neighbor discovery response (ND−RESP), in addition to the packets from anchors.
2) *connected*. The set of users is connected, but a given user can overhear only some of the others; while discovery through ranging exchanges is possible, the propagation



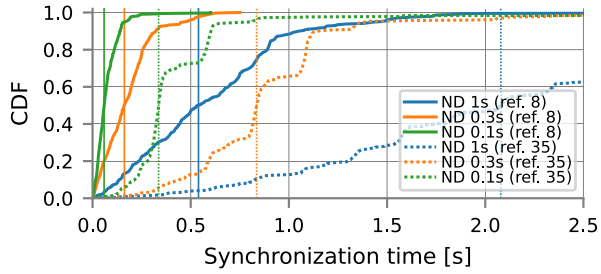
(a) *clique*.



(b) *connected*.

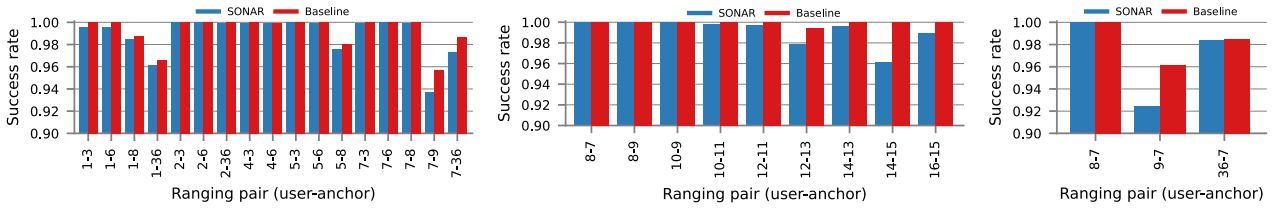


(c) *disconnected* (single anchor vs. multiple anchors between users).



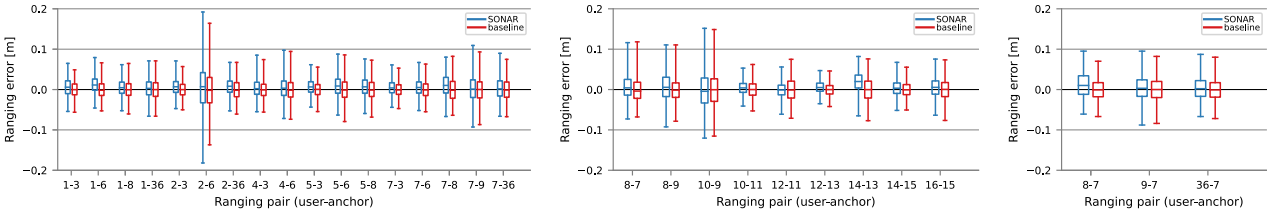
(d) *disconnected* (comparing ref. 8 and 35).

FIGURE 12. Synchronization time for all users across different scenarios, with varying discovery interval. Vertical lines represent the median values.



(a) *clique*. (b) *connected*. (c) *disconnected*.

FIGURE 13. **Comparison of the ranging success rate of SONAR and the baseline (T_{ND} = 1 s).**



(a) *clique*. (b) *connected*. (c) *disconnected*.

FIGURE 14. **Comparison of the distance estimation of SONAR and the baseline (T_{ND} = 1 s). Boxes represent the 25-75% percentile. Bars represent 0.1-99.9% of the data.**

of synchronization is not instantaneous. It potentially takes multiple handshakes spreading over multiple hops.

3) *disconnected*. The set of users is not connected, but at least one anchor exists that is shared among the separated groups of users; synchronization can only happen by means of neighbor discovery beacons (`ND-INIT`) or ranging responses (`RNG-RESP`) from the shared anchor.

By carefully selecting different nodes of the testbed, we can replicate the three scenarios above, and show that synchronization is achieved quickly regardless of the type of connectivity between users.

2) NODE SETUP

Figure 11 shows which nodes act as users and anchors in each scenario. For *clique*, all users are in a hall with no obstacles between them. Node 1 is the reference, i.e., the user with lowest ID that others must synchronize to. In *connected*, the limited transmission range creates a multi-hop synchronization structure. The `RNG` packets of node 8, the reference user assigned the lowest ID in this scenario, can reach 10, but not 12, 14 and 16. The *disconnected* scenario shows two separated groups of users, and only one node (7) that acts as an anchor. User 35 is the reference, in communication range with user 36. However, there is no link between them and the second group of users, making anchor 7 the *only* source of synchronization for users 8, 9, and 10. Users 10 and 35 also have weak `RNG` links to the anchor.

3) SYNCHRONIZATION TIME

We show CDFs for the synchronization time, i.e., the delay incurred before users align their slotframes to the reference. In principle, this metric can be computed as $t_{synch} - t_{sf}$, where t_{sf} and t_{synch} are, respectively, the theoretical start time of the slotframe and the synchronization timestamp of the non-reference user. However, we must also consider the random delays Δ_r and Δ_u applied by the reference and non-reference user at round start, yielding $t_{synch} - (t_{sf} + \max(\Delta_r, \Delta_u))$. By using max we account for the cases in which the reference user appears "late", i.e., $\Delta_r > \Delta_u$, as the non-reference user could not have synchronized before then.

First, we experiment with two users in range with each other and a single anchor. In this case, as expected, synchronization time is distributed between ∼0 ms and T_{ND}. Users do not transmit any packet until they discover at least one anchor, and neighbor discovery beacons are the only way for anchors to advertise their presence while in the isolated state. Once they do, the reference user discovers the first anchor and begins ranging. Its ranging polls are received by the other user(s), quickly reaching synchronization. Beyond this very basic case, we compute the synchronization time for our scenarios (*clique*, *connected*, *disconnected*) involving more anchors and users, with varying T_{ND} (1 s, 0.3 s, 0.1 s).

We proceed in order of scenario complexity, and present first the results in *clique* and *connected*. The synchronization time across all users is shown in Figure 12a and Figure 12b, respectively. In both cases, the CDF grows until the delay matches the neighbor discovery interval T_{ND}. In the case of *clique* with $T_{ND} = 0.1$ s, a slightly higher delay is seldom observed, which can be ascribed to occasional discovery failures due to contention. Instead, *connected* consistently shows longer tails of the distribution, due to the multi-hop topology which can sometimes delay the propagation of the reference when user 8 is not the first discovered. The impact of the multi-hop topology is clearly visible in Table 4, which

shows the per-user synchronization time. While *clique* users tend to all synchronize in a similar time, this metrics increases for *connected* users based on how far they are from the reference.

Next we analyze the results for the *disconnected* scenario, the most challenging of the three. Since the only way to propagate the synchronization information is through a single anchor, the delay can increase, especially for users that are not neighbors of the reference. As shown in the CDF (Figure 12c, "ref. 35") and per-user synchronization times (Table 4), users in the *disconnected* scenario typically need many ND intervals to synchronize. User 36 is consistently faster than the others, as it is close to the reference, user 35. However, if more than one anchor is available, synchronization time improves significantly for all users. We tested this variant, indicated with the label "many anchors", by enabling nodes 2 and 6 to also act as anchors. The improvement is also evidenced by the average per-user synchronization times. Further, if the reference is far from the anchor, its ND-RESP will often be superseded by one of the other concurrent replies whose signal is stronger due to proximity. This is the case when the reference user is node 35. Since multiple discovery attempts are often required, synchronization is delayed up to 4.85 s with $T_{ND} = 1$ s in the worst case. However, if we choose as reference the user closest to the anchor, node 8, synchronization is significantly faster (Figure 12d) since node 8 is discovered immediately. Table 4 ("ref. 8") reflects the improvements; as expected, node 36 no longer has an advantage w.r.t. the other users.

It is worth noting that the worst-case *disconnected* scenario is unlikely to appear in real deployments. The increased synchronization time depends on multiple factors affecting the system simultaneously: *(i)* multiple users entering anchor range at the same time, *(ii)* different sets of users that cannot communicate directly, *(iii)* a single anchor available for the communication between the sets of users, and finally *(iv)* the reference user being farther from the anchor w.r.t. the others. Even under these extremely unfavorable conditions, SONAR reliably achieves synchronization at the cost of a small increase in latency. If a long discovery interval is required for energy efficiency, but multiple users are expected to appear simultaneously (e.g., as in a swarm), SONAR anchors can be configured to transmit another ND-INIT following a successful discovery to speed up the process at a small energy cost.

TABLE 4. Mean and standard deviation of the synchronization time per user, ND interval, and scenario, in seconds.

T_{ND} (s)	Scenarios and user IDs			
	clique			
	2	4	5	7
0.1	0.04 ± 0.03	0.04 ± 0.04	0.05 ± 0.03	0.05 ± 0.04
0.3	0.08 ± 0.06	0.09 ± 0.06	0.09 ± 0.07	0.12 ± 0.06
1	0.29 ± 0.20	0.32 ± 0.22	0.34 ± 0.23	0.34 ± 0.23
	connected			
	10	12	14	16
0.1	0.06 ± 0.04	0.09 ± 0.05	0.13 ± 0.07	0.17 ± 0.09
0.3	0.14 ± 0.08	0.19 ± 0.11	0.25 ± 0.12	0.29 ± 0.12
1	0.39 ± 0.23	0.49 ± 0.24	0.57 ± 0.23	0.66 ± 0.24
	disconnected (ref. 35)			
	8	9	10	36
0.1	0.54 ± 0.85	0.54 ± 0.85	0.54 ± 0.85	0.31 ± 0.55
0.3	0.96 ± 0.49	0.96 ± 0.49	0.96 ± 0.49	0.74 ± 0.57
1	2.36 ± 0.89	2.11 ± 0.92	2.11 ± 0.92	1.83 ± 0.91
	disconnected (many anchors)			
	8	9	10	36
0.1	0.35 ± 0.33	0.35 ± 0.34	0.35 ± 0.33	0.16 ± 0.13
0.3	0.59 ± 0.78	0.60 ± 0.78	0.60 ± 0.78	0.40 ± 0.87
1	1.16 ± 1.12	1.17 ± 1.12	1.18 ± 1.12	0.83 ± 0.81
	disconnected (ref. 8)			
	9	10	35	36
0.1	0.07 ± 0.06	0.07 ± 0.06	0.09 ± 0.07	0.07 ± 0.06
0.3	0.16 ± 0.12	0.17 ± 0.13	0.22 ± 0.16	0.17 ± 0.12
1	0.55 ± 0.37	0.56 ± 0.37	0.76 ± 0.51	0.56 ± 0.36

4) RANGING SUCCESS RATE AND ERROR

After assessing the synchronization delay, we evaluate the reliability of ranging and the error in distance estimation. Because multiple users attempt to range with the same anchors, their packets could collide, or create interference in the CIR estimation. Frequent ranging failures or an increase in ranging error would then indicate that user transmissions often collide and/or interfere with each other. We compare against a baseline which schedules ranging exchanges in a round-robin fashion among nodes, with no chance of interference. For the baseline, the radio of the nodes is always on, but we do not observe significant difference in performance when following the same deep sleep schedule as SONAR.

For each pair of SONAR user and anchor, the ranging success rate is computed after two conditions are met: *(i)* the user has set the anchor as active for the first time (i.e., after mutual discovery), and *(ii)* the user has synchronized to the reference. Indeed, the notion of failed ranging is unclear before synchronization, when the shared schedule is not yet defined. before synchronization, when the shared schedule is not yeResults for $T_{ND} = 1$ s are reported in Figure 13. Similar success rates are obtained for the other discovery intervals. Overall, SONAR achieves a ranging success rate comparable to that of the baseline application.

Figure 14 shows, for each user-anchor pair in each connectivity scenario, the distance estimates of SONAR and the baseline, similar for all ranging pairs. For a numerical comparison, we consider the median and 99^{th} percentiles of the absolute ranging error in any user-anchor link. Across scenarios and ND intervals, the maximum discrepancy in performance between SONAR and the baseline is just 2.0 cm

for the median, and 2.8 cm for the 99^{th} percentile. This is a clear indication that the error distribution in SONAR is essentially unaffected by protocol operations, i.e., the established ranging schedule is free from interference.

VIII. CONCLUSION

UWB radios are generally considered too energy-hungry, therefore anchor infrastructures for ranging and/or localization are typically mains-powered. This allows anchors to keep their radio always on, and removes any concern about energy consumption. On the other hand, this approach increases installation costs, limits deployment flexibility, and even precludes UWB infrastructures entirely from environments with no access to the power grid.

This paper introduces SONAR, a novel protocol to support multiple users ranging with battery-powered anchors. Our design jointly provides efficient neighbor discovery, quick synchronization, and dynamic organization of the ranging schedule. By offloading radio operations to users, all these features come with minimal energy cost on the anchor side. SONAR is easily configurable in terms of discovery responsiveness and ranging rate, to adapt the protocol to different application needs. Indeed, we show that anchor lifetime can stretch to decades, making SONAR a practical solution for battery-powered UWB ranging and localization.

REFERENCES

[1] P. Corbalán, G. P. Picco, M. Coors, and V. Jain, "Self-localization of ultra-wideband anchors: From theory to practice," *IEEE Access*, vol. 11, pp. 29711–29725, 2023.

[2] M. Piavanini, L. Barbieri, M. Brambilla, M. Cerutti, S. Ercoli, A. Agili, and M. Nicoli, "A calibration method for antenna delay estimation and anchor self-localization in UWB systems," in *Proc. IEEE Int. Workshop Metrology Ind. 4.0 IoT (MetroInd4.0&IoT)*, Jun. 2022, pp. 173–177.

[3] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, "Self-calibration and collaborative localization for UWB positioning systems: A survey and future research directions," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–27, May 2022.

[4] F. Gottifredi and E. Varriale, "Navigation system for exploring and/or monitoring unknown and/or difficult environments," U.S. Patent 8 473 118, Jun. 25, 2013.

[5] J. P. Queralta, C. Martínez Almansa, F. Schiano, D. Floreano, and T. Westerlund, "UWB-based system for UAV localization in GNSS-denied environments: Characterization and dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4521–4528.

[6] N. Macoir, J. Bauwens, B. Jooris, B. Van Herbruggen, J. Rossey, J. Hoebeke, and E. De Poorter, "UWB localization with battery-powered wireless backbone for drone-based inventory management," *Sensors*, vol. 19, no. 3, p. 467, Jan. 2019.

[7] J. Bauwens, N. Macoir, S. Giannoulis, I. Moerman, and E. De Poorter, "UWB-MAC: MAC protocol for UWB localization using ultra-low power anchor nodes," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102637.

[8] E. Varriale, P. Corbalán, T. Istomin, and G. P. Picco, "PLaNS: An autonomous local navigation system," in *Proc. ION GNSS+, Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2018, pp. 1722–1727.

[9] N. Janicijevic, P. Corbalán, T. Istomin, G. P. Picco, and E. Varriale, "Small plans towards mars: Exploiting ultra-wideband for self-localizing rover navigation," in *Proc. Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*, 2018, pp. 199–200.

[10] J. Guo, X. Li, Z. Li, L. Hu, G. Yang, C. Zhao, D. Fairbairn, D. Watson, and M. Ge, "Multi-GNSS precise point positioning for precision agriculture," *Precis. Agricult.*, vol. 19, no. 5, pp. 895–911, Oct. 2018.

[11] D. Molteni, G. P. Picco, M. Trobinger, and D. Vecchia, "Cloves: A large-scale ultra-wideband testbed," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2022, pp. 808–809.

[12] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2020, 2020.

[13] A. Alarifi, A. S. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, p. 707, May 2016.

[14] J. Yang, B. Dong, and J. Wang, "VULoc: Accurate UWB localization for countless targets without synchronization," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 6, no. 3, pp. 1–25, Sep. 2022.

[15] J. Tiemann, F. Eckermann, and C. Wietfeld, "ATLAS—An open-source TDOA-based ultra-wideband localization system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–6.

[16] D. Vecchia, P. Corbalán, T. Istomin, and G. P. Picco, "TALLA: Large-scale TDoA localization with ultra-wideband radios," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.

[17] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3131–3137.

[18] P. Corbalán, G. P. Picco, and S. Palipana, "Chorus: UWB concurrent transmissions for GPS-like passive localization of countless targets," in *Proc. 18th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2019, pp. 133–144.

[19] M. Zhao, T. Chang, A. Arun, R. Ayyalasomayajula, C. Zhang, and D. Bharadia, "ULoc: Low-power, scalable and cm-accurate UWB-tag localization and tracking for indoor applications," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 3, pp. 1–31, Sep. 2021.

[20] B. Van Herbruggen, S. Luchie, R. Wilssens, and E. De Poorter, "Single anchor localization by combining UWB angle-of-arrival and two-way-ranging: An experimental evaluation of the DW3000," in *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, Jun. 2024, pp. 1–7.

[21] D. Vecchia, P. Corbalán, T. Istomin, and G. P. Picco, "Playing with fire: Exploring concurrent transmissions in ultra-wideband radios," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9.

[22] NASA. *Mars 2020: Perseverance Rover*. Accessed: May 27, 2025. [Online]. Available: https://science.nasa.gov/mission/mars-2020-perseverance/rover-components

[23] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Jun. 2004, pp. 455–462.

[24] P. Corbalán, T. Istomin, and G. P. Picco, "Poster: Enabling contiki on ultra-wideband radios," in *Proc. Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*, Feb. 2018, pp. 171–172.

[25] Qorvo. *DW1000 Data Sheet, Version 2.22*. Accessed: May 27, 2025. [Online]. Available: https://www.qorvo.com/products/p/DW1000

[26] Qorvo. *DWM1001 Battery Lifetime Estimation*. Accessed: May 27, 2025. [Online]. Available: https://forum.qorvo.com/t/dwm1001-battery-lifetime-estimator/4244

DAVIDE VECCHIA received the M.Sc. degree in computer science and the Ph.D. degree in information and communication technology from the University of Trento, Italy, in 2018 and 2022, respectively. He is a Post-Doctoral Researcher at the Department of Information Engineering and Computer Science, University of Trento. His research interests include non-line-of-sight detection and error mitigation for wearable ultra-wideband (UWB) devices and energy-efficient protocol design based on concurrent transmissions. His research focuses on UWB communication and localization in large-scale multi-hop networks, for which he received the Best Paper Awards at IPSN in 2023 and IPIN in 2019.

PABLO CORBALÁN received the M.Eng. degree in telecommunication engineering from Miguel Hernández University, Spain, in 2014, and the Ph.D. degree in information and communication technology from the University of Trento, Italy, in 2020. He is currently a Software System Engineer at NXP Semiconductors, where he works on the design of UWB solutions for mobile systems and actively contributes to UWB standardization efforts. His main research interests are in low-power wireless communications and localization. His research has received several awards, including the Best Paper Award at EWSN in 2018 and IPIN in 2019 and the Best Poster Award at EWSN in 2016.

TIMOFEI ISTOMIN made early contributions to this work during his tenure as a Researcher at the University of Trento. He is currently an Engineer at Qorvo, where he focuses on ultra-wideband (UWB) products. His professional expertise spans networked embedded systems, distributed computing, low-power wireless communication/localization systems, and software engineering.

GIAN PIETRO PICCO (Senior Member, IEEE) is a Professor at the Department of Information Engineering and Computer Science, University of Trento, Italy. His research spans the fields of software engineering, middleware, and networking, and is currently oriented towards wireless sensor networks, the Internet of Things and cyber-physical systems, mobile computing, and large-scale distributed systems. The research performed in his group combines theoretical study and in-field validation in real-world applications, and has led to several awards, including the Most Influential Paper at ICSE'07 (for a paper published a decade earlier) and the Best Paper Awards at IPSN in 2009, 2011, 2015, and 2023, PerCom in 2012, EWSN in 2018, and IPIN 2019. He has served as an Associate Editor for IEEE TRANSACTIONS ON SOFTWARE ENGINEERING (TSE) and the *Journal of Pervasive and Mobile Computing*. He was also the founding Editor-in-Chief of *ACM Transactions on the Internet of Things* (TIOT). He is an Associate Editor for *ACM Transactions on Sensor Networks* (TOSN).

ENRICO VARRIALE joined Thales Alenia Space Italia (TASI) in 2007, contributing to navigation projects, including the development of the time reference system for the Galileo navigation constellation. He also worked on industrial research and development projects focusing on time and frequency metrology, distributed computing systems, and algorithms for the monitoring of critical infrastructures and autonomous navigation solutions. Currently, as part of the Chief Technical Office at TASI, he works on coordinating research and development activities in navigation, Earth observation, and telecommunications. He continues engaging in research focused on distributed computing and networking in orbit, as well as the development and integration of quantum technologies for sensing, metrology, and communications.