6Fit-A-Part: A Protocol for Physical Distancing on a Custom Wearable Device

Yifeng Cao, Ashutosh Dhekne, Mostafa Ammar
Georgia Institute of Technology, {ycao361, dhekne}@gatech.edu, ammar@cc.gatech.edu

***Abstract*—The coronavirus pandemic is altering our way of life. As more establishments open, there is an expectation that people will follow physical distancing guidelines. The implementation, however, is poor; just putting up warning signs appealing the general public to keep a distance of 6 feet from others is hardly enough. In this paper we consider the design of a wearable device that raises an alarm if another similar device is detected within a set distance. It uses off-the-shelf ultra-wideband radio technology for real-time, accurate distance estimation from others in the vicinity. We design an one-to-all ranging protocol that is able to accurately estimate distance to neighboring devices and warn the user if the distance falls below a certain established threshold within a short time. The device must compensate for human occlusions and avoid unnecessary warnings when physical barriers exist between devices. We implement and evaluate our protocol in a small testbed with custom prototype hardware as well as in simulation. Our ranging protocol is capable of performing up to 10 distance measurements per second, while avoiding packet collisions. The overall percentage of rangings completed is around 65% in a 10-node network, and the distance accuracy is around 20cm even with frequent human occlusions. We believe this prototype will provide the first steps to ensure physical distancing in various real-world settings.**

***Index Terms*—physical distancing, UWB, ranging protocol, LOS NLOS, ultra-wideband, social distancing, occlusions**

I. Introduction

This paper develops a distance measurement protocol running on a custom wearable device that aids maintaining physical distancing. It is envisioned to help, for example, teams of doctors and nurses in a hospital, coworkers in restaurants or warehouses, and even shoppers in a mall or grocery store. But first, we begin by motivating the need for such a device during the current COVID-19 pandemic and beyond. A virus is a submicroscopic infectious agent that replicates only inside the living cells of an organism [44]. This need of a living host cell leads to an effective strategy in slowing a viral epidemic's spread: prevent the virus from finding new people to infect. Physically preventing a microorganism from infecting a new victim is a set of non-pharmaceutical interventions that have proved effective throughout history [32], [35], [40], [45], by preventing a surge of patients that would inundate the healthcare system [12], [28]. The CDC suggests [20] keeping a 6 feet distance to thwart the possibility of infections.

Strategies such as stay-at-home encourage physical distancing, yet, many essential establishments must remain open and functioning. The expectation from people working at and visiting these essential establishments, such as hospitals, pharmacies, warehouses, grocery stores etc., is that they will practice physical distancing. Can we ensure that *people remain at a safe distance from each other*, by actively warning them when they are too close? In this work, we consider how to employ technology to serve as a friendly reminder—a nudge to maintain safe distance. This paper explores the **technical hindrances** of creating such technology, though a more thorough system evaluation should include psychological and sociological studies as well.

978-1-7281-6992-7/20/$31.00 ©2020 IEEE

More specifically, we design and prototype a wearable device that warns the user if it detects another similar device in close proximity. When co-workers must maintain physical distancing (e.g., in hospitals, warehouses, etc.), employees can clip this device to pockets, much like ID cards (Fig. 1(a)). Medical professionals can be grouped into teams enabling intra-team contact without impediment, while cautioning when inter-team exposure might occur (Fig. 1(b)). At grocery stores where the general public needs physical distancing, this device can be clamped to grocery carts or baskets (Fig. 1(c)). We call this device, *6Fit-a-Part*, since it is a "part" that can fit onto clothing (Fig. 1(d)) or equipment. 6Fit-a-Part constantly monitors the surroundings for potential "personal space" violations. Using ultra-wideband radios and a modified wireless ranging protocol, it performs distance measurements with its peers, and achieves high accuracy in real-world environments.

Of course, there is no novelty in simply performing wireless distance measurements; wireless localization already does that. So creating such a device and the associated protocol design might seem trivial. However, several challenges emerge when we start considering the problem carefully:

1) Devices may arbitrarily enter and exit from each other's vicinity, and need to measure distance with all nearby devices, without a central authority. This results in hundreds of messages being exchanged per second **crippling existing protocols**.
2) Such devices are useful only if they work reliably every time. It is therefore important to **minimize false-positive and -negative rates**—no nuisance, or missed warning.
3) Without the right context, distance measurements can often be misleading. Two shoppers in different aisles need not be warned even if close-by, whereas a person body-blocking the device may cause imperfect distance estimates. An effective solution **must differentiate such contexts**.

Wireless distance measurement can be based off a variety of techniques, from WiFi channel state information (CSI), Bluetooth signal strength, to acoustic ranging. Each of these techniques can be easily employed on the ubiquitous smart-

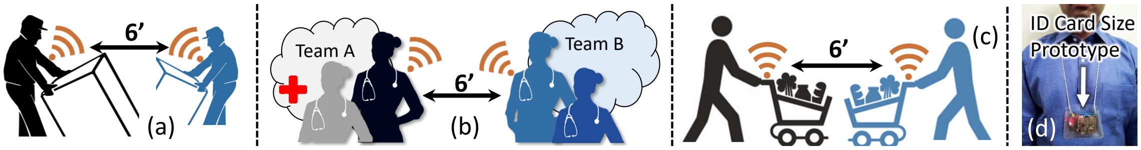


Fig. 1. 6Fit-a-Part applications: (a) co-workers at warehouses, restaurants, (b) healthcare teams at hospitals, (c) shoppers at stores; (d) wearable prototype

phone. It is, therefore, natural to wonder if a specialized electronic device is really necessary. However, in real-world environments, such as inside a grocery store, or in a hospital, each of these methods face significant hurdles. Constant sonic chirps can be irritating to people, while inaudible ultra-sound can raise health concerns [47]. Real-world environments are cluttered and cause dynamically changing multipath, rendering WiFi CSI based techniques unsuitable due to their small 20-80MHz bandwidths. Bluetooth signal strength is unreliable in the best of conditions, and the store environment further exacerbates the problem. To tackle these real-world issues, we propose to use ultra-wideband (UWB) radios, a fast-growing technology (iPhone 11, Samsung Galaxy Note are equipped with UWB radios) which can obtain nanosecond receive timestamps owing to their 1GHz bandwidths. Moreover, UWB's fine-grained channel impulse response increases multipath resolution; path-lengths larger than $\approx 30\,cm$ can be clearly differentiated. Our core contributions are:

1) Developing a **peer-peer wireless ranging protocol** to efficiently determine numerous pair-wise distances between near-by devices (measuring about 10 distances per second).
2) Analyzing the wireless channel impulse response to differentiate clear line-of-sight measurements from human-occluded and barrier-occluded ones, **correcting distance estimates** (median accuracy less than 20cm).
3) Designing and prototyping an ID-card size device on which to instantiate the ranging protocol. The device is low power ($53\,mA$ transmit current), lightweight ($\approx 50gms - 100gms$), small ($\approx 8cm \times 5cm$), and inexpensive ($\approx \$50$).

The rest of the paper is organized as follows. We start with an overview of the 6Fit-a-Part system with a brief description of each module. We then discuss details of the peer-peer ranging protocol including design-decisions rationale. We then discuss the occlusion detection and correction process, followed by device implementation details, and evaluation of the complete system. We then discuss prior work in this area, followed by a short discussion of future work, and conclusion.

II. System Overview

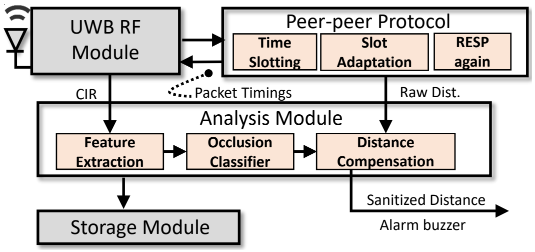


Fig. 2. 6Fit-a-Part System Architecture

Fig. 2 shows the overall 6Fit-a-Part architecture. The *front-end UWB RF (radio frequency) module* is responsible for sending and receiving ultra-wideband packets. It records precise nanosecond-level timestamps of outgoing and incoming packets, and also extracts the channel impulse response (CIR). The timing information is used by the *ranging module* to run the 6Fit-a-Part peer-peer ranging protocol. It decides how many slots to allocate for responders in every round of the protocol, maintains slot timings, and enables a quick retry mechanism for colliding responders. In conjunction with the UWB RF front-end, the ranging module produces a raw estimate of the distance between peers. This distance estimate and the CIR from the UWB RF module is processed inside the *analysis module* to infer the context of the measurement. Specifically, the analysis module extracts features from CIR and classifies the obtained distance measurement into line-of-sight (LOS), human occlusion, or barrier occlusion, (together called NLOS), and compensates for distance errors, if required. The output of the analysis module is this sanitized distance and an alarm trigger, if necessary. Finally, all decision attributes are stored by the storage unit for offline analysis and improvements.

III. Protocol Design

6Fit-a-Part is based on the core primitive of wireless distance estimation between devices. However, existing ranging protocols assume a small number of "anchors" perform these measurements [18], [22], [25]. Naively stretching these to many-to-many distance measurements would be inefficient, yielding low ranging rates. We first describe the existing standard protocol and then discuss our contributions in designing a faster ranging protocol, in detail.

A. Wireless Ranging Primer

A robust distance estimate can be obtained by measuring the time of flight (ToF) from a transmitter to a receiver, and then multiplying ToF with the speed of light. ToF-based ranging places stringent demands on the ToF precision, since, even a small error in ToF, amplifies significantly when multiplied with the speed of light ($\approx 3 \times 10^8 m/s$) leading to large distance errors. This problem is further exacerbated when clocks on the two devices are not synchronized and can drift independently of each other. UWB transceivers with a large wireless bandwidth allow precise ToF estimation. To minimize effects of unsynchronized clocks and clock drifts, ranging protocols, such as the symmetric two-way ranging (TWR) protocol [25] have been devised.

Fig. 3 shows this standard TWR protocol between devices A and B. The device A initiates distance measurement by sending a POLL packet to another device B. On reception of a POLL, B records the precise reception time, then *turns into a transmitter*, and sends a RESP packet back to A. At this stage, one *round trip* has been completed. A, on its part records the reception time and sends the FINAL packet back to B. At this

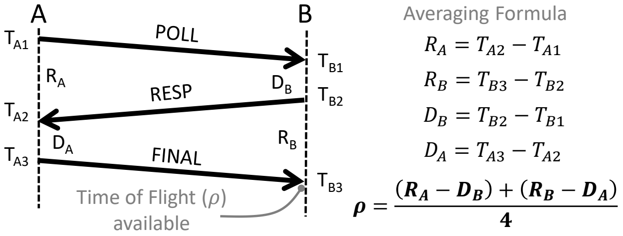


Fig. 3. The IEEE802.15.4 [25] TWR ranging protocol with averaging equations to compute propagation delay.

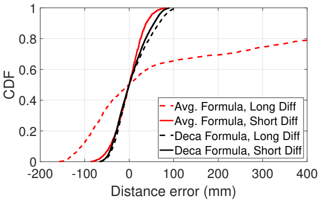


Fig. 4. Effect of different turn-around times on ranging precision using standard averaging [25] vs using Decawave technique from [33].

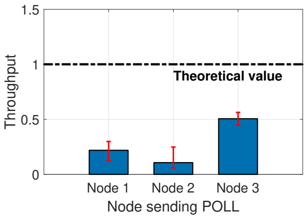


Fig. 5. Naive TWR: Collisions result in sub-optimal number of rangings per unit time. Capture effect causes unfair number of rangings.

point, two round trips (denoted by R_x) and two turn-arounds (denoted by D_x) have occurred. Device B now has enough information to calculate its distance from A, by averaging the differences in the various transmission and reception times (see the right side of fig. 3 for the averaging formula). Using relative instead of absolute time, TWR eliminates clock offset between A and B. Furthermore, it can be shown that the error caused by the clock drifts at the two devices is minimized by minimizing the difference in the two turn around times.

However, this requirement of minimizing the difference in the turn-around time limits the scope of protocol modifications. Fig. 4 shows the performance degradation of the averaging formula with increasing difference in turn-around times. Instead, an improved empirical formula has been proposed in [33], we call it the Deca formula, making the ranging precision independent of the difference in turn-around times. In this formulation, ToF (ρ) is given by:

$$\rho = \frac{R_A \cdot R_B - D_A \cdot D_B}{R_A + R_B + D_A + D_B}. \quad (1)$$

Proof of this is trivial and we refer the reader to [33], [34] for details. The key strength of this formula is in the robustness to different turn-around times, as demonstrated mathematically in [34]. Compared to the averaging formula, eq. (1)'s robustness can be seen in fig. 4. We utilize this greater flexibility in designing our modified one-to-many ranging protocol.

B. Peer-Peer Ranging

The above TWR protocol is suitable only for finding distance between *two designated* nodes. 6Fit-a-Part, on the other hand, requires all devices to range with any other device in its vicinity, without knowing their identity *a priori*. We therefore need a modified ranging protocol, that at a high level, satisfies the following requirements: (1) Effectiveness: Each node measures the distance from any and all devices that are close-by. (2) Efficiency: 6Fit-a-Part should minimize packet collisions within the network of nodes, which would otherwise cause performance degradation in a multi-node scenario. (3) Realtime behavior: 6Fit-a-Part should maximize the number of distance measurements per second to quickly raise an alarm when two devices are too close.

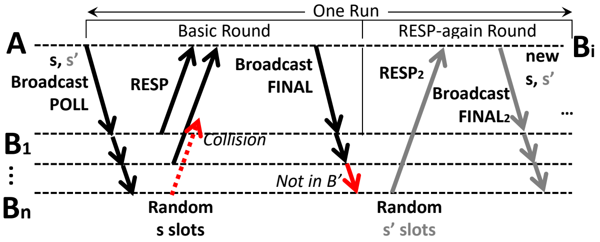


Fig. 6. The 6Fit-a-Part one-to-many ranging protocol initiated by A.

The first requirement begets the question: *Who is close-by?* A naive way to approach this question is by sending out a broadcast request as a discovery process. Performed by all devices in the vicinity, every device would know of the entire network's membership. This discovery step could be followed by a series of node-wise TWR operations. However, for a network with n mobile nodes, such a protocol would first require n discovery messages, and then all nodes will have to perform the TWR protocol followed by a distance report message with $n-1$ other nodes, for a total of $n + 4n(n-1)/2 = 2n^2 - n$ messages. With frequent message exchanges, uncontrolled pair-wise transmissions incur significant collisions. Fig. 5 shows the impact of frequent collisions: the number of usable rangings falls short of the theoretical maximum. Further, it is difficult to guarantee fairness, where all nodes perform equal number of rangings, due to potential capture effect.

We propose below a protocol to satisfy the requirements in such a multi-node scenario. Overall, the 6Fit-a-Part protocol can be summarised as follows: (i) A more efficient message exchange scheme that reduces the total number of message exchanges to n^2+n, ($\approx 2\times$ improvement); (ii) A slotted time division multiple access scheme with real-time slot adaptation to reduce RESP collisions; and (iii) A two-step scheme allowing RESP re-transmission while a collision occurs. The core benefit in the 6Fit-a-Part protocol comes from the exploitation of the broadcast nature of wireless channels, which is further enhanced by piggybacking information pertaining to multiple nodes on a single wireless packet. Next, we describe the proposed 6Fit-a-Part protocol, but relegate justification of the design choices to section III-C.

Fig. 6 shows the flow of messages in 6Fit-a-Part protocol. Since every device is equivalent and each of them runs exactly the same protocol, there is no master device or central controller. Each device runs a simple state machine shown in fig. 7. For ease of explanation, we consider the protocol from the perspective of one particular device, called A and first follow the "POLL-Path" in fig. 7. A has (n) neighbors $B_1, B_2, \cdots, B_n$ at a given instance. These nodes follow the "RESP-Path" in fig. 7.

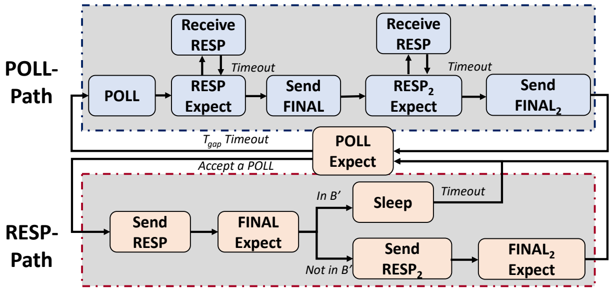
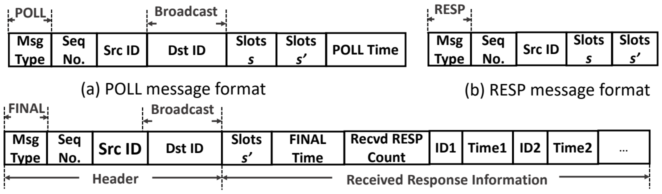


Fig. 7. 6Fit-a-Part state machine. Node A follows POLL-path, while B_i follow RESP-path, in tandem. Roles switch after every run.



(a) POLL message format

(b) RESP message format

(c) FINAL message format

Fig. 8. Message Formats of POLL, RESP, and FINAL

Basic round: The protocol starts with A sending a *broadcast* POLL packet, (packet structures are shown in fig. 8). Within the POLL, A declares the number of slots, s, available to any B_i for sending RESPs. Every slot is defined as a non-overlapping short period of time starting from the reception of the POLL packet. After sending the POLL, A changes to being a receiver and records every response it receives. This process will last for s slots. It then waits an additional short period σ as a guard interval. On reception of the broadcast POLL, each receiving device $B_{1..n}$ must randomly choose one among the s slots, and send its RESP packet in that slot. Since the propagation delay is much smaller than the transmission delay, POLL arrives almost simultaneously at all B_i, ensuring the non-overlapping property of slots at different B_i nodes. After sending the RESP, B_i enters the FINAL expecting state, waiting for the FINAL from A.

There are three possibilities for each slot that A has given: (i) no responding device chose that particular slot, (ii) only one of $B_{1..n}$ devices transmitted in that slot, (iii) two or more devices send RESP in the same slot (a collision occurred), which may produce an error in the reception. After dutifully waiting for s time, A builds a set $\{B'_{1..m}\}$ which contains the type (ii) RESPs above. A then assembles the reception times and node IDs in $\{B'\}$ as well as other information that will be used by the B' devices to calculate their own distance from A, (message format is shown in fig. 8). Meanwhile, based on the number of empty or collision slots observed, A *dynamically adapts the distributed slots* s *in every turn* to efficiently trade-off between minimizing the time delay and reducing the collisions. In general, higher occupation rate indicates that a larger s is required to avoid collisions. More details on the slot adaptation are in section III-C. On receiving the FINAL from A, every B_i checks if $B_i \in B'$. If it is a member, it calculates the distance from A. Otherwise, B_i participates in the respond-again (RESP-again) round described below.

Respond-again round: While the basic round reduces the collisions through slotting and slot adaptation, a fraction of packets are still lost due to collisions. For instance, as shown in fig. 9, when there are 5 nodes, the packet reception ratio is approximately 80% even when distributing slots with 4× the number of nodes—increasing number of slots provides diminishing returns. To mitigate this issue, we propose the respond-again scheme. Specifically, for nodes who are not in B', it is allowed to send another RESP randomly choosing among further s' slots after A's FINAL is received. The number s' is given in the FINAL packets (and also in POLL packets to avoid certain corner cases) sent by A. This allows A to receive a new set of RESPs and send another FINAL to these nodes denoted as B''. Nodes already present in B' do not participate in this round, and instead sleep for a short time saving power. A's POLL to FINAL_2 is called *one run*.

At this point, a total of $|B' \cup B''| = n'$ devices have calculated their distance from A. Revisiting the state diagram, A has completed POLL-path, while the B_i's have completed the RESP-path. A itself, despite having initiated the process, does not receive any distance estimates. Next, A starts a timer T_{gap} and enters the POLL-expect state of RESP-path during which it waits for a POLL from another node. While it waits, A may go back to sending a POLL with an increasing probability p_{poll} as time passes. If it does receive a POLL, it freezes T_{gap} and enters the RESP-path flipping its role and behaving like a B node. The T_{gap} timer remains frozen until the current full round (both basic and respond-again parts) finish, after which T_{gap} starts running freely once again. This process repeats until T_{gap} runs out: once again node behavior flips, A enters the POLL-path and initializes a new POLL.

C. Design Decisions

None of the choices in this protocol are arbitrary. In this section, we discuss the motivation for the design choices made in the 6Fit-a-Part peer-to-peer protocol, through comprehensive simulation and experiments. The focus is on 6Fit-a-Part's robustness and efficiency in a multi-node setting.

*1) **Benefit of the two-round RESP strategy**:* We show that adding more slots to the basic round, instead of having a two-round RESP, does not reduce collisions. Adding slots provides more nodes a contention-free chance of sending the RESP packet to A. However, every slot consumes valuable time that reduces the ranging rate. Hence, we must carefully choose the number of slots made available to B_i. Suppose the proportion of nodes not experiencing a contention is p_{clean} with s slots. The proportion p_{clean} is a function of both the number of available slots s and the number of nodes n:

$$p_{clean} = F(s, n). \tag{2}$$

Fig. 9 shows the percentage of contention-free nodes given a certain number of available slots. Increasing the number of slots available to the whole network produces diminishing

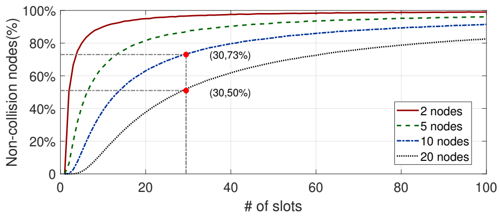


Fig. 9. Increasing number of slots provides diminishing returns.

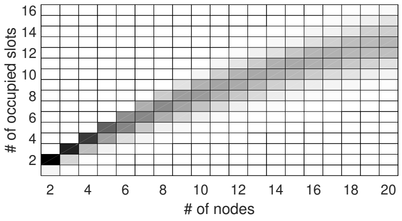


Fig. 10. Relationship between number of nodes and number of occupied slots.

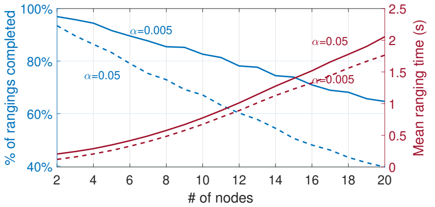


Fig. 11. Trade-off: A slower p_{POLL} rate reduces collisions, but increases wasted time.

returns. In a 20-node network, 50% nodes (i.e. 10 nodes) will enjoy contention free slots if 30 slots are available. However, even if an *additional* 30 slots are made available to the whole network, only 70% nodes will have enjoyed contention free slots. Instead, if we first present only 30 slots, and then present another 30 slots to the previously un-represented nodes, a total of 85% nodes will have sent a RESP (10/20 in the basic round, 7/10 in the respond-again round). Therefore, a 2-stage strategy is beneficial over a single-stage strategy.

*2) **Guidance for the choice of** s:* The number of available slots is dynamically adapted to avoid bandwidth wastage. Our choice of the number s during every round has a significant impact on the overall packet reception ratio. Fig. 9 provides an intuition about s, but we will formalize it here. Ideally, we want to choose s based on our best guess of the number of nodes currently present in the vicinity (network size), which can be inferred from the slot occupancy in the previous RESP-expect state. As an example, fixing the number of maximum slots to 20, fig. 10 shows the probability of the number of occupied slots as the number of nodes in the network changes. Evidently, the space of the most likely network size is small once the number of occupied-slots is known. 6Fit-a-Part uses fig. 10 as a look-up table for network size inference $\mathcal{I}$. Given that s_{occ} slots were occupied in the previous run, a particular row in fig. 10 will be relevant. Pick the cell in that row which has the highest probability, resolving ties by preferring the higher number of nodes. This is the best estimate of the number of nodes in the network. Then, with estimated number of nodes, 6Fit-a-Part uses fig. 9 to obtain the value of s such that a preset percentile p_{clean} of those nodes will receive a contention free slot. In our protocol, we empirically set it to be higher than 80% between the basic and RESP-again rounds. This choice of s is calculated at *every run* independently by every node. Given our lookup table approach, we use the inverse function of eq. (2) to update the slots:

$$s = F^{-1}(p_{clean}, \mathcal{I}(s_{prev}, s_{occ})). \tag{3}$$

*3) **Choice of** s':* The choice of s' is tightly coupled with the choice of s and how exactly s was chosen. Overall, the goal is to achieve ranging with higher than a certain percentage of nodes. Therefore, if s was chosen at the 50%, s' will have to be larger than if s was chosen at 80%, for example. However, the choice of s' can be made, without loss of generality, at the same time that s is chosen. This is particularly beneficial in ensuring that overhearing neighbors remain frozen for the entire duration of the current run, as will become clear in section III-C7, where we discuss multiple collision domains.

*4) **Choice of inter-POLL time** T_{gap}:* After sending the FINAL$_2$ message, the node will wait for T_{gap} before initializing another POLL. Since distance is calculated at the nodes who *receive* the FINAL, sending a POLL is altruistic. Smaller T_{gap} brings little benefit, but increases the POLL collision rate. POLL-collision occurs when two (or more) nodes A_1 and A_2 send a POLL almost simultaneously. In that case, both nodes will miss the other's POLL and enter the RESP expected state making no progress. In addition, too frequent POLLs would also drain the battery quickly. Therefore, from A's perspective, it is better-off sending the *least* number of POLL messages, and giving others a chance to send their POLLs. On the other hand, since all devices are peers, A cannot entirely stop sending POLLs, otherwise no rangings will occur in the network. 6Fit-a-Part sets the T_{gap} based on the following principles:

1) Incessant POLLs from A do not add new information.
2) T_{gap} should allow other nodes to send their POLLs.
3) T_{gap} should minimize POLL collisions.
4) T_{gap} should enable conserving battery.

Drawing from these principles, the value of the inter-POLL gap T_{gap} is probabilistic. A tunable parameter α dictates the rate at which p_{poll} increases with the time since last POLL. Fig. 11 shows α's impact on PRC and MRT, providing guidance about selecting the design parameter α. In addition, to avoid fast battery drain, A does not POLL for a short time ($150ms$ in our experiment). If A is alone in the network, it will wait for at least $150ms$ before sending another POLL. After that, A would send out a POLL with slowly increasing probability p_{POLL} when it is in the POLL-expect state. Any time A is responding to another node's POLL message, A will be in any state in RESP-path except POLL-expect state and will not initiate its own POLL.

*5) **Choice of freezing inter-POLL timer**:* 6Fit-a-Part freezes the timer T_{gap} when it accepts a POLL from others. Had we not frozen the timer, when another node B_1 sends a POLL, A will send a RESP but still keep counting down on its T_{gap}. At the same time, other nodes such as B_i which might have a counter value close to A will also count down. Suppose A reaches zero first, but cannot transmit because it has to wait for the current B_1 run to complete. While A is waiting on zero, B_i will also reach 0 and wait for B_1 to finish. At that instant, A and B_i's POLL packets will collide. By introducing the timer freeze, we ensure that only empty slots are counted reducing the collision probability to only those nodes who were at the same count simultaneously. This idea

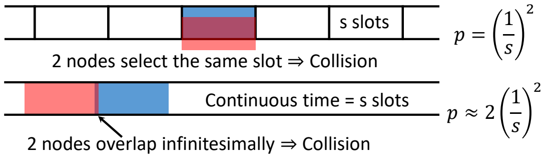

s slots
$p = \left(\frac{1}{s}\right)^2$
2 nodes select the same slot ⇒ Collision
Continuous time = s slots
$p \approx 2\left(\frac{1}{s}\right)^2$
2 nodes overlap infinitesimally ⇒ Collision

Fig. 12. Slotted access reduces collisions, compared to unslotted access.

can create a logjam where, once two nodes start colliding, they will *always* collide. A small random backoff at the T_{gap} prevents this. Nodes respect other POLLs and freeze during that time, staying on the small backoff.

6) ***Slots vs continuous time for RESP back-off****:* The 6Fit-a-Part protocol suggests all B nodes pick a random slot from $1..s$ and transmit in that slot. Instead, it is also possible to treat time as continuous and allow nodes to choose at much finer time-scales. Slots are naturally more efficient at avoiding collisions (see comparison in fig. 12). The only drawback of slotted back-off is the requirement of a globally synchronized time. In any other wireless system, this would be a major hurdle. Fortunately, the reception of the broadcast POLL packet and accurate time-stamping available on UWB devices already provides a uniform way to synchronize all B nodes[1]. Since *node-synchronization comes for free*, slotted back-off behavior is naturally preferable for 6Fit-a-Part.

7) ***Multiple collision domains****:* While the wireless channel is broadcast, it does not spread out to infinity. In a large space, it will be easily possible to have more than one collision domain; nodes too far to hear A's POLL packets. If such a node, say C is far enough to not have any interactions with A's neighbors (B's), then C can POLL simultaneously with A. In general, however, C may be far enough to not hear the POLL, but still hear B's RESPs. In such a case, C may unintentionally interfere with B. To avoid this interference, if C detects an unsolicited RESP packet during its POLL-expect state, C freezes for the amount of time that $A - B$ communication is expected to continue. Every RESP packet contains s, and s', to help C determine the freezing time.

IV. Occlusion Aware Distance Compensation

A. Challenges Due to Occlusions

In dynamically changing environments, as the ones we expect at hospitals, warehouses, and grocery stores, the 6Fit-a-Part peer-peer ranging protocol is essential to quickly detect presence of others in close vicinity. However, the protocol alone is insufficient; our ultimate aim is to create an alarm system to warn of potential safe distance violations. Whether or not to raise an alarm for a specific distance can depend on the context. For example, two devices in two different rooms must not trigger an alarm even if they are indeed close-by, since they are separated by a physical wall; a *false positive* case. On the contrary, two devices occluded by humans, may appear to be farther away than they actually are, since the direct signal path is blocked, while a stray reflection (longer path) might arrive at the receiver, providing a false sense of security, since the devices will register a longer-than-actual distance; a *false negative* case. Avoiding such problems needs us to handle distances differently based on the context.

1) Detecting Physical Barriers: The general problem of detecting barriers is not simple to solve in its entirety. Wireless signals, unlike light, are not completely blocked by brick walls, wooden panels, and other obstacles. Meaning, a device pair may still be in wireless range even when they are separated by a brick wall. We will obtain a range from such a pair, and our task is to look at this measurement, use any auxiliary information available, and determine if this measurement corresponds to LOS link, or a physical barrier was present.

2) Detecting Body Blocking: Signal absorption by the human body creates unique challenges for distance estimation since the direct signal is attenuated significantly, causing diffraction or a reflection from a nearby object to be misidentified as the first path. This results in an overestimation of the actual distance. Fig. 13(a), borrowed from [19] (and slightly modified for explanation here), closely demonstrates this effect, including occlusions at different angles. UWB device A, is surrounded by several other UWB devices Bs with some of them occluded due to body-blocking (a person standing between A and B_2, B_3). The distance measurements made by A with each of these devices are plotted as green or red dots in fig. 13(a). The devices that have clear line-of-sight with A obtain consistent measurements, whereas the occluded devices within about 30°cone behind the person (B_2, B_3) obtain wildly fluctuating estimates (the red streaks). This toy example illustrates the issues 6Fit-a-Part will face when occluded by people—a common scenario in any public space.

We ask: *Is it possible to distinguish between human occlusions and physical barriers? Is it possible to obtain accurate distance estimates despite human occlusions?*

B. Classification Opportunities

To answer these questions, we need to first understand the process by which UWB devices decide the packet reception time and other auxiliary information that might be useful. The UWB receiver performs a cross correlation of incoming physical layer preamble with a known pseudo-random sequence. This correlated signal is then convolved with the known transmission pulse to obtain a high-bandwidth CIR, which captures the signal strengths and time delays between echoes. The resolution of this “impulse” is limited by the signal's bandwidth, but since UWB devices use a large bandwidth, we obtain a few nanosecond resolution in the CIR[2] with every CIR point associated with a timestamp. An internal leading edge (LDE) detection algorithm analyzes this signal, and determines the first instance when the signal amplitude exceeds a previously set LDE threshold. That timestamp is reported as the packet reception time.

Ideally, this receive timestamp should be accurate irrespective of whether the devices currently have a clear LOS between them or not. However, environmental factors can lower this

[1] The slight difference in arrivals due to propagation delay is a few nanoseconds and can be safely absorbed within slot guard-bands

[2] A 1GHz bandwidth allows 1ns resolution. Practically, path-lengths $3-4\,ns$ apart can be clearly distinguished. $1ns \approx 1foot$ difference in path lengths.

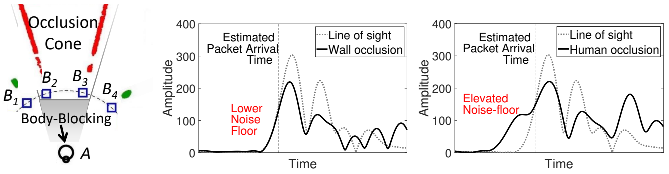


Fig. 13. (a) Body-blocking causes erroneous distance measurements [19], (b) Line-of-sight (LOS) vs Wall occluded CIR (c) LOS vs human occluded CIR.

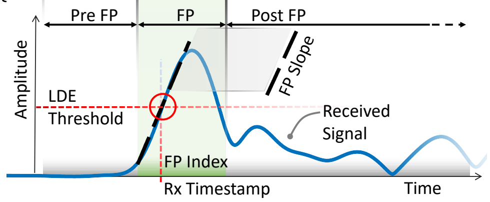


Fig. 14. Observation of various signal components enables feature selection.

accuracy. The LDE threshold is calibrated to provide consistent results during line-of-sight operations. Walls and other barriers attenuate wireless signals, so for the same distance, a barrier-occluded signal's first path will have lower amplitude than their line-of-sight counterparts. Fig. 13(b) shows a CIR with wall occlusions. In contrast, when the first path is body-occluded, the attenuation can be so severe that its amplitude can lie below the LDE threshold. Such a first peak will be missed completely (fig. 13(c)). We believe careful analysis of the CIR presents an opportunity to correct distance estimates.

This knowledge of the packet reception process leads us to two key observations to detect and classify occlusions.

Observation 1: Occluded signals (through walls, or humans), are attenuated more than LoS signals at the same distance.

Observation 2: Human occlusion distorts the shape of the peak with closely-coupled multipath, causing the wrong peak to be selected as first path.

Next, we discuss our classification and distance correction approach based on these key observations.

C. Classification And Distance Correction Methodology

Our goal is to create a module that takes a CIR and measured distance as input, and first classifies it into LOS, wall-occlusion, or human-occlusion. For the final case, the module also computes a distance correction. Several CIR features will prove useful in this endeavor: Fig. 14 provides a glimpse into these features. Three regions are marked: (1) the region before the first path (Pre FP), (2) the region during the first path (FP), (3) the region after the first path (post FP). One important feature is that if we have missed capturing the first path, then the "Pre FP" region contains attenuated signals, and not noise, elevating the signal strength in that region. If no line-of-sight path exists in this CIR, then we expect the "Post FP" region to have similar signal strengths as the "FP" region since all paths are only reflected paths. A weak or slowly rising first path pulse will have a smaller slope than a quickly rising LOS first path. Each of these properties affect the computed "Rx Timestamp", which we intend to correct. Our features will be based off these "if-this-then-that" observations.

One might be tempted to approach the occlusion classification problem using a fine-grained deep learning model. However, 6Fit-a-Part is based on a small embedded microcontroller with just 32KB of RAM and without Internet connectivity. These severe resource constraints preclude the use of sophisticated deep learning algorithms. However, since we use UWB radios, we do obtain a fine-resolution multipath profile of the environment providing us opportunities to simplify the classification algorithm. Therefore, based on the "if-this-then-that" nature of the solution described above, we argue that a decision tree is a suitable model for both classification and distance compensation. Although distance compensation should ideally be a continuous value, and a regression tree only provides a small set of values, we still find it beneficial. Table I describes the features used for fitting the two trees. Signal attenuation (x9) due to obstructions is the most important feature to distinguish between occlusions and LOS. The FP Slope is indicative of distance errors for human occlusions (fig. 13(c)). Some features incorporate the raw distance (*) using Friis transmission formula [39] to avoid over-fitting.

Label	Feature	Importance (DST/RGT)
x1	FP to Pre-FP power ratio (P_{fp}/P_{pre})	1.69% / 1.67%
x2	First Peak Index	2.34% / 0
x3	Slope of best-fit line at FP	22.9% / 51.13%
x4	Pre-FP Power (P_{pre})	0 / 2.34%
x5	FP Power (P_{fp})	6.97% / 6.33%
x6	Raw distance measurement*	NA / NA
x7	FP to post-FP power ratio (P_{fp}/P_{post})	11.02% / 14.0%
x8	Pulse shape (Correlation coefficiency)	4.24% / 0
x9	Receive signal power (P_{recv})	50.84% / 24.53%

TABLE I
FEATURES USED IN CLASSIFICATION AND REGRESSION TREE

A classification tree (DST: 50% split, 3% error) and a regression tree (RGT: 25-75% train-test split, $2.8, 7.7, 18.8\,cm$ resubstitution error at $50, 75, 90\%ile$) were trained on Table I's features (fig. 15) using data collected in three environments including: low multipath (open driveway, medium multipath (large living room), and high multipath (busy kitchen area). Fig. 16 demonstrates the effectiveness of distance compensation in the human occlusion case. CIR and ranging data was collected at multiple distances (letters A, B, C, D in fig. 16) with and without human occlusion. The LOS CDFs are shown as solid-black lines, whereas red lines show the corresponding raw human-occluded ranges. Post compensation, the human-occlusion CDFs (blue dotted lines) closely mimic LOS ranges.

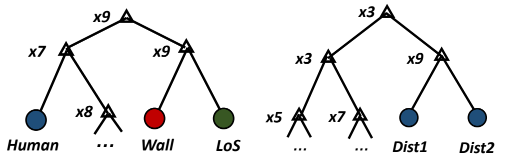


Fig. 15. (a) Classification DST, (b) Distance compensation RGT.

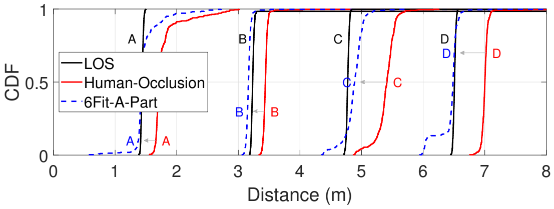


Fig. 16. Occluded distance estimates are improved by 6Fit-a-Part.

In summary, every range measurement is first classified into LOS, wall-occlusions, or human-occlusions, and only compensate the distance for human-occlusions.

V. Implementation Details

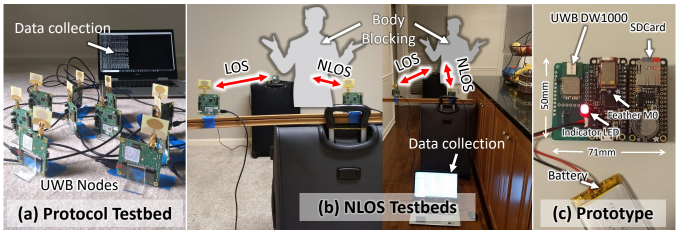


Fig. 17. (a) Protocol evaluation testbed: 10 UWB nodes (b) Occlusion testbed: UWB nodes on movable luggage (c) 6Fit-a-Part hardware prototype.

The 6Fit-a-Part protocol has been completely implemented, in C, on a UWB development platform, Decawave TREK1000 [6]. All protocol evaluation in section VI has been performed on this platform operated at $4\,GHz$ center frequency, with $6.8\,Mbps$ data rate. A 10 node testbed used for evaluating the protocol is shown in fig. 17(a). Distance calculations occur on the TREK1000 nodes. Some of these nodes are connected over a USB-serial port to a Dell Inspiron laptop, with an Intel Core i7 microprocessor, running Matlab R2019a on Windows 10. The laptop collects distance estimates, timing information, and node IDs, for detailed evaluation.

In the NLOS testbeds (fig. 17(b)), the laptop also collects CIR data from the nodes, useful for obtaining LOS, human occlusion, and wall occlusion data in various environments. The distance compensation RGT is run in Matlab on the laptop (post processing). However, we have deliberately kept the prediction logic simple so that it can be ported to the embedded platform and run in real-time.

Finally, fig. 17(c) shows our hardware prototype that integrates the Decawave DW1000 chip [7] ($18) with Adafruit Feather M0 [2] which runs the Atmel ATSAMD21G18 ARM Cortex M0 processor [4] ($20), with a 32KB RAM, and 256KB of program memory. We *over-provision* this hardware platform using Adalogger Featherwing [3] ($9) which supports writing CIR information to an SDCard ($3). The platform is powered using a 1200mAh battery ($9) and is expected to last about 10 hours [7]. This research prototype runs the 6Fit-a-Part protocol, including distance compensation, and stores CIR data for debugging. The final product, sans the logger module, will cost $50. The software and hardware design files are available at: https://github.com/Whisper-Cao/6Fit-a-part

VI. Evaluation

A. Overall Network Efficiency

In a distributed peer-peer protocol with no central coordinator, packet collisions cause reduction in efficiency as the number of nodes in the network increases. Using the protocol evaluation testbed, we inspect the percentage of rangings completed (PRC). Fig. 18(a) shows the change in PRC as we sequentially increase the number of nodes in the network. To collect enough data packets, the network is run long enough for a minimum of 500 rangings from every node. Our implementation of the 6Fit-a-Part protocol supports a 10-node network with PRC at about 65%. Note that the PRC is different from, and usually much lower than, the network's packet reception rate, since for a ranging to complete, *the whole set* POLL-RESP-FINAL must be received. The 6Fit-a-Part protocol involves slotting for responses, random backoff, and a RESP-again strategy. As a baseline, we implement another protocol, called arbitrary TWR (A-TWR), where a node receiving a POLL immediately responds. The only modification in A-TWR to the standard TWR is broadcasting POLLs and FINALs, and including all receive times in the FINAL packets. In this case, multiple responses collide, and typically the initiator only receives one of those responses (a manifestation of the capture effect). As a result, A-TWR's PRC falls quickly with increase in network size.

The PRC directly affects the rate of acquiring distance measurements from others, which we investigate next: How long would it take for any node to receive one ranging from all of its peers? We average the time between successive FINAL packets from the same node obtained over all the rangings received, which approximates the network's mean ranging time (MRT). Fig. 18(b) compares the increase in the MRT for 6Fit-a-Part vs that for A-TWR as the network grows from 2 to 10 nodes. 6Fit-a-Part has a 10-node MRT of about 1second (on an average, every node receives ranges from 9 other nodes in 1 second.) In comparison, A-TWR performs extremely poorly. In fact, beyond 6 nodes A-TWR rarely completes ranging with all nodes. Our results for the 10-node network are promising; in the real-world this translates to 10 shopping carts in vicinity, or teams of 10 medical staff in close proximity receiving distance estimates from everyone, every second.

B. Micro-benchmarks

1) Tail Latency: The MRT shown in section VI-A fails to capture the worst case behavior of the network even with error-bars. However, in our use-case, not receiving a range from one of the nodes for a long time means *no alarm is raised* even when such a node violates the 6-feet rule. Fig. 19 shows an inverted CDF of the ranging time for networks of various sizes. Compared to A-TWR, 6Fit-a-Part protocol manages to keep the tail latency low. The 99% (95%) MRT for 6Fit-a-Part is $3.6\,seconds$ ($2.2\,seconds$) for a 10-node network, whereas it is $24.7\,seconds$ ($13.3\,seconds$) for the A-TWR protocol.

2) Fairness: In a peer-peer network, being fair entails making a similar number of distance measurements with every other node. We demonstrate 6Fit-a-Part's fairness in fig. 20 and compare it with A-TWR for a modest 6-node network[3]. Every node in 6Fit-a-Part performs approximately the same number of total distance measurements, and each node ranges with all other nodes approximately the same number of times.

3) Selection of Slotting Threshold: As mentioned before, slot adaptation is based on the estimation of nodes in the network. Yet, we rely on presetting the expected proportion of successful distance measurements p_{clean}, used in the basic round. We evaluate the percentage of ranging that completed

[3] Every node must collect fairness data, logistically limiting to fewer nodes.

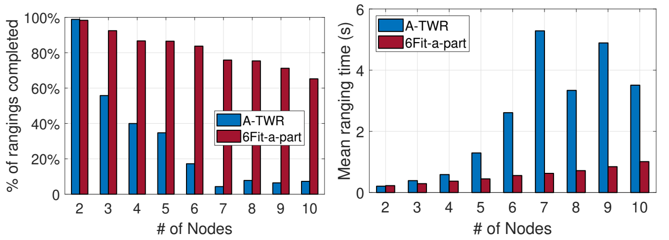


Fig. 18. 6Fit-a-Part is robust to increasing network size. Compared across two dimensions: (a) PRC (b) MRT. A-TWR erodes on collisions.

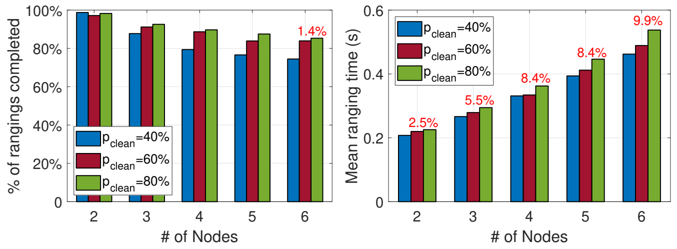


Fig. 21. The effect of slot adaptation on: (a) PRC, (b) MRT. Increasing p_{clean} gives diminishing returns, but increases ranging time.

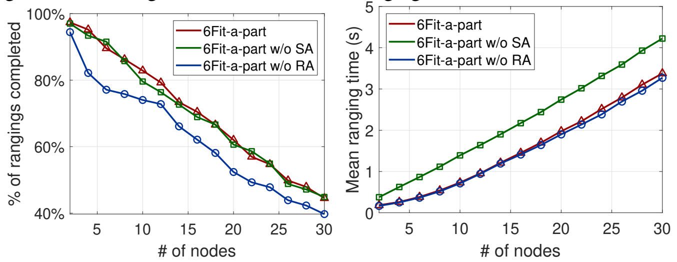


Fig. 22. SA = slot adaptation, RA = RESP-again. The effect of number of nodes on: (a) PRC (b) MRT.

when different p_{clean} values are employed and the results are shown in fig. 21(a). As expected, offering more slots provides diminishing returns. And the flip side of a higher p_{clean} can be observed from the increase in ranging time in fig. 21(b). Note that while the % of rangings completed is probabilistic, the increase in mean ranging time is deterministic—the trend is therefore easier to observe in fig. 21(b).

4) Effect of Slot Adaptation, RESP-again: Evaluation of slot adaptation scheme and turning the RESP-again on/off is difficult on a real test-bed. Therefore we evaluate these parameters in a trace-driven simulation environment. Similar to fig. 18, we show the PRC and MRT for the simulation in fig. 22. When testing without slot-adaptation, we set the number of slots to the maximum available slots (the upper bound of what 6Fit-a-Part protcol can choose). RESP-again scheme is left on for this case. Observe in fig. 22(a) that 6Fit-a-Part protocol achieves almost the same PRC as when using fixed maximum slots. The cost we pay by fixing the number of slots is an increased MRT (see green line in fig. 22). If we turn off the RESP-again scheme, PRC reduces but we do not gain too much on the MRT dimension. This shows that the overhead of RESP-again scheme is minimal. We are thus better-off using both RESP-again and slot adaptation.

5) Dynamic Network Size: Fig. 23 show the network's PRC and MRT as the number of nodes in the network is changed. It shows that the network adapts to the increased network size in real-time and restores prior performance on removal of nodes. Fig. 23 also shows the observed variability in PRC and MRT metrics, precluding the need for error-bars in fig. 18.

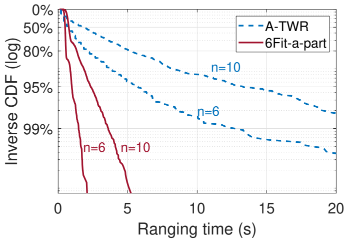


Fig. 19. Tail latency for 6Fit-a-Part is much better than A-TWR.

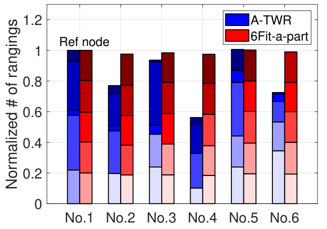


Fig. 20. 6Fit-a-Part achieves fairness: ranges equally with all nodes. A-TWR rangings vary significantly.

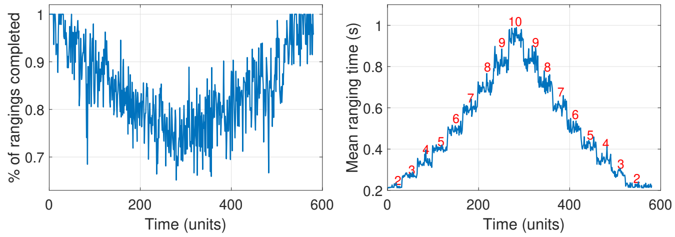


Fig. 23. (a) PRC decreases, (b) MRT increases, with increasing network size; removing nodes restores both PRC and MRT to previous values.

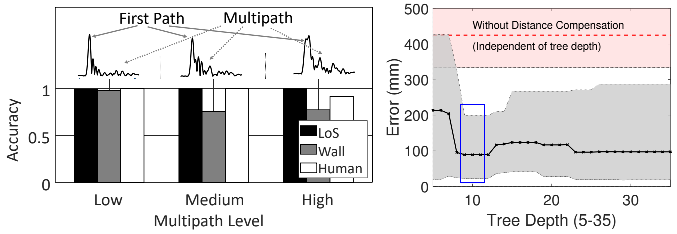


Fig. 24. (a) Accuracy of occlusion detection. (b) Distance compensation accuracy vs selected regression tree depth.

6) Occlusion Classification: We test 6Fit-a-Part's classification accuracy in three environments with increasing multipath: an open area, a living room with sparse furniture, and a kitchen with severe multipath. Fig. 24(a) shows representative CIRs in these environments. Experiments are performed at distances $1-4\,m$ between two UWB devices ($1\,m$ granularity) for line-of-sight, wall occlusion, and human occlusion in these environments. Fig. 24(a) shows that 6Fit-a-Part accurately classifies in low-multipath environments. The accuracy degrades with increasing multipath, yet the overall accuracy is 93% (85%) for medium- (high-) multipath scenarios.

7) Human-Occlusion Distance Compensation: We evaluate the regression tree described in section IV-C by varying tree depths in fig. 24(b). The median distance error, without compensation, and independent of tree depth, is shown as the red dashed line (shaded pink region shows the $10^{th} - 90^{th}$ percentile error). As we change the tree depth, 6Fit-a-Part achieves different median errors shown by the black line (shaded gray region is the $10^{th} - 90^{th}$ percentile errors). We select the tree depth that leads to a low median and low 90% around $20\,cm$, (marked rectangle). Fig. 25(a) demonstrates 6Fit-a-Part's distance compensation for measurements made by a cart moving from $5\,m$ to about $1\,m$ from another cart, while being pushed by a person (human occlusion). Efficacy of the distance compensation is shown in fig. 25(b) for other environmental scenarios; errors experienced before compensation (red lines) are reduced after compensation (black lines).

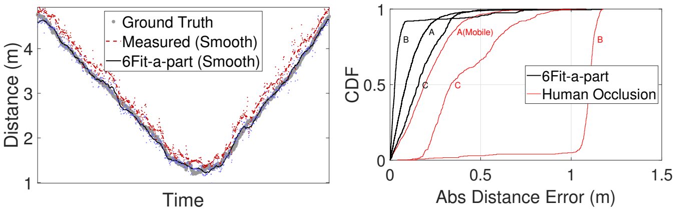


Fig. 25. (a) Measured and 6Fit-a-Part compensated distance for a moving cart, (b) 6Fit-a-Part reduces distance error in various human-occlusion scenarios.

VII. Related Work

Ranging approaches: Ranging-oriented RF sensing has been investigated for its use in diverse applications such as localization, navigation and SLAM [21], [30]. Common ranging approaches include WiFi [27], [37], RFID [38], [43] and mmWave [13]. However, bandwidth limitations, degrade WiFi-based ranging precision. On the other hand, mmWave suffers from fast attenuation. As an alternative, 802.15.4 based ultra-wideband radios [25] strikes a good balance between ranging precision and longer range with $1\,GHz$ bandwidth. UWB radios are already present in iPhone 11 and Samsung Galaxy Note—a trend that we expect to grow. Existing UWB ranging works include [24], [29], [31], [41], which employ TWR as the protocol backbone. 6Fit-a-Part extends TWR to a peer-peer protocol, enhancing utility of any UWB hardware.

Multi-user ranging: Collision is a fundamental problem in wireless networks. In spite of extensive efforts on the design of collision-aware protocols [10], [11], [46], collisions in TWR protocol have not been explored until recently. [15] proposes using the time difference among peaks in CIR to estimate the distance of all responders, which is based on the observation that the highest power packet is still likely to be decoded. However, it is inapplicable in the real-world because of its inability to identify the responder. To solve the identification problem, SnapLoc [23] assigns an individual delay in the nanosecond range to each node. Chorus [16] uses both the pulse shape and response position to encode the identification of a node. However, both SnapLoc and Chorus cannot support dynamic network membership, and peer-peer measurements. Instead, 6Fit-a-Part solves the multi-user ranging in the MAC layer by dynamically allocating slots to each responder. Compared to the existing works, 6Fit-a-Part is more flexible in that (i) 6Fit-a-Part stays stable even as nodes enter and leave and (ii) every node switches its role periodically. Hence every node in the network can perform ranging with others.

Contact Tracing: Multiple COVID-19 contact tracing applications exist [1], [5], [8], [9]. At their core, these apps create a mobile phone proximity database, which is consulted when a person is diagnosed positive. Others who might have come in close contact with this person are warned. Instead of such contact-backtracking *after potential exposure*, 6Fit-a-Part aims to *prevent* all unnecessary contact with strangers in real-time. Thus, 6Fit-a-Part is distinctly different from contact tracing, though both can simultaneously co-exist.

LOS/NLOS classification: Classification of obtained signals into LOS/NLOS buckets has been previously performed in various contexts [17], [36], [42], [48]. The general idea is to exploit the CSI to infer channel conditions, thereby determine whether a signal is LOS. Recently, machine learning based LOS/NLOS is proposed to improve the classification accuracy. For instance, [14] takes a series CSI estimates, and runs a RNN structure for classification. Since 6Fit-a-Part is designed for lightweight embedded devices, running deep learning algorithms is unrealistic. 6Fit-a-Part uses simple decision trees and bases its classification and distance compensation on features from the channel impulse response.

VIII. Discussion and Future Work

Further improvements would be required to transform 6Fit-a-Part into a product, though this work tackles the fundamental problems of multi-user ranging and occlusion-aware sensing.

Multiple Collision Domains: Our current evaluation of 6Fit-a-Part focuses only on a single collision domain i.e, all nodes can hear each other. When overlapping collision domains exist, the measurement frequency degrades for the overlapping regions. We leave this exploration for future work.

Generality of Decision Tree Approach: While 6Fit-a-Part is designed, implemented, and evaluated in a small testbed mimicking the real-world, we do not claim that occlusion-aware compensation is perfect. Recent works in wireless sensing [26] shine some light on achieving generality in such cases through machine learning approaches. Adapting those for our tiny embedded device would be an interesting future challenge. section VI tests 6Fit-a-Part using dry-walls as physical barriers. 6Fit-a-Part does not detect thin acrylic glass barriers or single plywood sheets.

Beyond COVID-19: Physical distancing, as a principle, is an effective prevention strategy against propagation of any infectious disease. Continued use of 6Fit-a-Part at hospitals and clinics can effectively keep infections localized and prevent outbreaks. As a general peer-peer distance measurement system, 6Fit-a-Part offers benefits beyond COVID-19.

IX. Conclusion

This paper shows the technical feasibility of performing peer-peer ranging to obtain pairwise distances in real-world environments with multiple peers in the vicinity. We have developed a small wearable device that can be worn as an armband or a badge, or attached to shopping carts, etc. The benefit of deploying 6Fit-a-Part is immediate even if *one* grocery store or hospital adopts the idea. We believe 6Fit-a-Part will provide the right tools today, and continue to remain valuable in the future, enabling a fundamental distance measurement capability, with utility beyond COVID-19.

X. Acknowledgments

We thank the anonymous reviewers for their comments and suggestions. We thank our shepherd, Dr. Lili Qiu, for guiding us during the final phase of this paper. This material is based upon work supported by the NSF under Grant No. 2031868.

References

[1] Aarogyasetu. https://www.mygov.in/aarogya-setu-app/.

[2] Adafruit feather m0 datasheet. https://cdn-shop.adafruit.com/product-files/2267/MDBT40-P256R.pdf.

[3] Adalogger featherwing m0. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-adalogger-featherwing.pdf?timestamp=1590359077.

[4] Atmel atsamd21g18 armcortex m0 processor. http://ww1.microchip.com/downloads/en/DeviceDoc/SAM_D21_DA1_Family_DataSheet_DS40001882F.pdf.

[5] Covidwise. https://www.vdh.virginia.gov/covidwise/.

[6] Decawave trek1000 datasheet. https://www.decawave.com/wp-content/uploads/2018/09/trek1000_user_manual.pdf.

[7] Dw1000 chip datasheet. https://www.decawave.com/sites/default/files/resources/dwm1000-datasheet-v1.3.pdf.

[8] Privacy-preserving contact tracing. https://www.apple.com/covid19/contacttracing.

[9] Nadeem Ahmed, Regio A Michelin, Wanli Xue, Sushmita Ruj, Robert Malaney, Salil S Kanhere, Aruna Seneviratne, Wen Hu, Helge Janicke, and Sanjay Jha. A survey of covid-19 contact tracing apps. *arXiv preprint arXiv:2006.10306*, 2020.

[10] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. Macaw: a media access protocol for wireless lan's. *ACM SIGCOMM Computer Communication Review*, 24(4):212–225, 1994.

[11] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320, 2006.

[12] Peter Caley, David J Philp, and Kevin McCracken. Quantifying social distancing arising from pandemic influenza. *Journal of the Royal Society Interface*, 5(23):631–639, 2008.

[13] Christian Carlowitz, Martin Vossiek, Axel Strobel, and Frank Ellinger. Precise ranging and simultaneous high speed data transfer using mm-wave regenerative active backscatter tags. In *2013 IEEE International Conference on RFID (RFID)*, pages 253–260. IEEE, 2013.

[14] Jeong-Sik Choi, Woong-Hee Lee, Jae-Hyun Lee, Jong-Ho Lee, and Seong-Cheol Kim. Deep learning based nlos identification with commodity wlan devices. *IEEE Transactions on Vehicular Technology*, 67(4):3295–3303, 2017.

[15] Pablo Corbalán and Gian Pietro Picco. Concurrent ranging in ultra-wideband radios: Experimental evidence, challenges, and opportunities. In *EWSN*, pages 55–66, 2018.

[16] Pablo Corbalán, Gian Pietro Picco, and Sameera Palipana. Chorus: Uwb concurrent transmissions for gps-like passive localization of countless targets. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 133–144. IEEE, 2019.

[17] William PL Cully, Simon L Cotton, William G Scanlon, and JB McQuiston. Body shadowing mitigation using differentiated los/nlos channel models for rssi-based monte carlo personnel localization. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 694–698. IEEE, 2012.

[18] Ashutosh Dhekne, Ayon Chakraborty, Karthikeyan Sundaresan, and Sampath Rangarajan. Trackio: tracking first responders inside-out. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pages 751–764, 2019.

[19] Ashutosh Dhekne, Umberto J Ravaioli, and Romit Roy Choudhury. P2ploc: Peer-to-peer localization of fast-moving entities. *Computer*, 51(10):94–98, 2018.

[20] Center for Disease Control and Prevention. Social distancing, quarantine, and isolation, 2020. "[Online; published 4/4/2020]".

[21] Sebastian Gansemer, Uwe Großmann, and Syuzanna Hakobyan. Rssi-based euclidean distance algorithm for indoor positioning adapted for the use in dynamically changing wlan environments and multi-level buildings. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–6. IEEE, 2010.

[22] Mahanth Gowda, Ashutosh Dhekne, Sheng Shen, Romit Roy Choudhury, Lei Yang, Suresh Golwalkar, and Alexander Essanian. Bringing iot to sports analytics. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 499–513, 2017.

[23] Bernhard Groβwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. Snaploc: an ultra-fast uwb-based indoor localization system for an unlimited number of tags. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 61–72. IEEE, 2019.

[24] Bernhard Großwindhager, Michael Rath, Josef Kulmer, Mustafa S Bakr, Carlo Alberto Boano, Klaus Witrisal, and Kay Römer. Salma: Uwb-based single-anchor localization system using multipath assistance. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 132–144, 2018.

[25] Jose A. Gutierrez, Edgar H. Callaway, and Raymond L. Barrett. *IEEE 802.15.4 Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks*. IEEE, 2003.

[26] Unsoo Ha, Junshan Leng, Alaa Khaddaj, and Fadel Adib. Food and liquid sensing in practical environments using rfids. In *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pages 1083–1100, 2020.

[27] Mohamed Ibrahim, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Richard Howard, Bo Yu, and Fan Bai. Verification: Accuracy evaluation of wifi fine time measurements on an open platform. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 417–427, 2018.

[28] Tom Jefferson, Ruth Foxlee, Chris Del Mar, Liz Dooley, Eliana Ferroni, Bill Hewak, Adi Prabhala, Sree Nair, and Alex Rivetti. Physical interventions to interrupt or reduce the spread of respiratory viruses: systematic review. *Bmj*, 336(7635):77–80, 2008.

[29] Benjamin Kempke, Pat Pannuto, Bradford Campbell, and Prabal Dutta. Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 137–149, 2016.

[30] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 99–112, 2012.

[31] Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. A robot self-localization system using one-way ultra-wideband communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3137. IEEE, 2015.

[32] Helen Marshall, Philip Ryan, Don Roberton, Jackie Street, and Maureen Watson. Pandemic influenza and community preparedness. *American Journal of Public Health*, 99(S2):S365–S371, 2009.

[33] Michael McLaughlin and Billy Verso. Asymmetric double-sided two-way ranging in an uwb communication system, February 14 2016.

[34] Dries Neirynck, Eric Luk, and Michael McLaughlin. An alternative double-sided two-way ranging method. In *2016 13th workshop on positioning, navigation and communications (WPNC)*, pages 1–4. IEEE, 2016.

[35] Karl David Patterson. *Pandemic influenza, 1700-1900: a study in historical epidemiology*. Rowman & Littlefield Totowa, NJ, USA:, 1986.

[36] Amanda Prorok, Phillip Tomé, and Alcherio Martinoli. Accommodation of nlos for ultra-wideband tdoa localization in single-and multi-robot systems. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–9. IEEE, 2011.

[37] Maurizio Rea, Aymen Fakhreddine, Domenico Giustiniano, and Vincent Lenders. Filtering noisy 802.11 time-of-flight ranging measurements from commoditized wifi radios. *IEEE/ACM Transactions on Networking*, 25(4):2514–2527, 2017.

[38] Jihoon Ryoo and Samir R Das. Phase-based ranging of rfid tags with applications to shopping cart localization. In *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 245–249, 2015.

[39] Joseph A Shaw. Radiometry and the friis transmission equation. *American journal of physics*, 81(1):33–37, 2013.

[40] Richard Smith. Social measures may control pandemic flu better than drugs and vaccines, 2007.

[41] Janis Tiemann, Fabian Eckermann, and Christian Wietfeld. Atlas-an open-source tdoa-based ultra-wideband localization system. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, 2016.

[42] Alexander V Vorobyov and Alexander Georgievic Yarovoy. Human body impact on uwb antenna radiation. *Progress In Electromagnetics Research*, 22:259–269, 2012.

[43] Jue Wang and Dina Katabi. Dude, where's my card? rfid positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 51–62, 2013.

[44] Wikipedia contributors. Virus — Wikipedia, the free encyclopedia, 2020. [Online; accessed 27-April-2020].

[45] Wikisource. Catholic encyclopedia (1913)/leprosy — wikisource,, 2013. [Online; accessed 27-April-2020].

[46] Kaixin Xu, Mario Gerla, and Sang Bae. Effectiveness of rts/cts handshake in ieee 802.11 based ad hoc networks. *Ad hoc networks*, 1(1):107–123, 2003.

[47] Chen Yan, Kevin Fu, and Wenyuan Xu. On cuba, diplomats, ultrasound, and intermodulation distortion. *Computers in biology and medicine*, 104:250–266, 2019.

[48] Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan, Haibin Cai, Yunhao Liu, and Lionel M Ni. Wifi-based indoor line-of-sight identification. *IEEE Transactions on Wireless Communications*, 14(11):6125–6136, 2015.