Optimal Swarm Ranging Protocol for Dynamic and Dense Ultra-Wideband Networks

Yunxi Hou*, Feng Shan*🖂, Wangxiao Mao†, Jiangpeng Liu†
Ye Liu*, Wenjia Wu*, Runqun Xiong*, Junzhou Luo*‡
* School of Computer Science and Engineering, Southeast University, Jiangsu, China.
† College of Software Engineering, Southeast University, Jiangsu, China.
‡ School of Computing and Artificial Intelligence, Fuyao University of Science and Technology, Fujian, China.
{houyunxi, shanfeng, maowangxiao, liujiangpeng, yeliu, wjwu, rxiong, jluo}@seu.edu.cn, 🖂 Correspondence: F. Shan

***Abstract*—Nowadays, as mobile robots and devices become smaller and lighter, forming them into swarms to collaboratively complete tasks has become an important research area. The previously introduced Ultra-Wideband (UWB) Swarm Ranging (*SRv1*) protocol pioneered simultaneous data transmission and ranging. However, it suffers from performance degradation in large-scale robot or device swarms.**

This paper introduces Swarm Ranging 2.0, a fundamentally redesigned and theoretically optimal protocol, which pushes the DS-TWR method to its theoretical limit, maximizing the number of distance calculations. Firstly, we propose a novel *compensatory ranging* method, enabling additional ranging for dynamic swarms. Next, we analyze the primary packet loss scenarios and redesign the ranging message and ranging table (data structure) to achieve robust ranging. Subsequently, to cope with complex combinations of packet loss and inconsistent frequency, we model the new protocol using a state machine. Theoretical analysis further proves its optimality. We implement the protocol on Crazyflie 2.1 drones equipped with DW3000 UWB transceivers. Experiments with 25 drones show a 47.8% improvement over *SRv1* and over 300% improvement compared to standard UWB protocol, demonstrating the protocol's scalability and effectiveness in real-world swarm deployments. The protocol is open-sourced at https://github.com/SEU-NetSI/crazyflie-firmware.

***Index Terms*—Ultra-Wideband, Swarm Ranging, Algorithm Design, Protocol Design**

I. Introduction

Nowadays, with the rapid development of microelectronics technology and semiconductor manufacturing, various mobile robots and devices are evolving towards miniaturization and lightweight design. For instance, in April 2025, Shiwei *et al.* [1] introduced a miniature wireless amphibious land-air robot measuring 9 centimeters in length and weighing 25 grams. In July 2024, Shen *et al.* [2] introduced the CoulombFly, a micro aerial vehicle less than 5 grams. Bitcraze released the micro-drone Crazyflie 2.1 in February 2019 [3], which is only the size of a palm and weighs 27 grams.

Compared to traditional medium and large robots and devices, micro robots and devices exhibit advantages such as low cost, small size, and high maneuverability. These features make them particularly suitable for operation in narrow or complex environments. However, the limited onboard computing and power capacity significantly constrains the range and complexity of tasks executable by individual micro robots. Consequently, forming swarms of micro robots or devices to work collaboratively is key to overcoming these limitations and enhancing task execution efficiency. This has become a focal point of current research [4], [5], [6], [7].

Due to the characteristics of micro robots or devices swarms, such as large numbers, high dynamics, and small distance, real-time neighbor distance sensing, *e.g.,* ranging, is fundamental for ensuring the reliability, stability, and safety among micro robots and devices. Shan *et al.* [8], [9] innovatively proposed the ultra-wideband (UWB) *swarm ranging* protocol that utilizes the broadcast nature of wireless channels to simultaneously perform data transmission and distance ranging for neighbors within the swarm. We reference it as *Swarm Ranging version 1.0* protocol, or *SRv1* for short.

We tried to implement the *SRv1* protocol in a much larger microrobot swarm than that in the original work [8], [9]. During our experiment, we identified two key drawbacks that motivated us to design a new protocol that better supports dynamic and dense swarms.

Motivation 1. We found that the larger the swarm size, the more likely ranging messages may be lost due to wireless channel conflicts. In a 25-robot swarm, 22.3% of messages were lost when ranging messages were transmitted to others every 60 ms. *SRv1* handles this situation so poorly that the loss of a single ranging message causes multiple ranging opportunities missed. Specifically, only 49.6% successful ranging rate compared to 77.7% success messages reception rate by *SRv1*. **Core idea 1**. Although message loss causes incomplete timestamps for distance computing in a traditional way, we discovered a previously unexplored way to compute an additional ranging distance using these incomplete timestamps. We call this method *compensatory ranging*. By carefully redesigning the ranging messages and the ranging table (data structure), the impact of message loss is minimized.

Motivation 2. We attempted to analyze the theoretical limits of the *SRv1* protocol in maximizing the number of ranging operations. We found that in the design of *SRv1*, issues such as ranging message loss and inconsistent frequencies are handled on a case-by-case basis, making the protocol complex, hard to scale, and difficult to analyze. **Core idea 2**. We therefore completely re-designed the protocol by adopting a state machine-based approach. With the help of a state machine, the new protocol is proven to be optimal in maximizing the number of distance calculations and approaches the theoretical limits.

Therefore, this paper aims to redesign and upgrade the original protocol toward the *Swarm Ranging version 2.0* protocol, referred to as *SRv2*. Special attention is paid to handling packet[1] loss and inconsistent ranging frequencies, as these are common in dynamic and dense swarm environments.

The contributions of this paper are summarized as follows.

1) We discovered a previously unexplored way to compute an additional ranging distance using incomplete timestamps, and introduce the *compensatory ranging* method. Together with the careful redesign of ranging messages and ranging tables, the *SRv2* protocol not only minimizes the impact of ranging message loss, but also enables more effective ranging under inconsistent ranging frequency.
2) The *SRv2* protocol's design is simple, systematic and theoretically optimal. We adopt a state machine-based design and offer a strict proof of the protocol's optimality, demonstrating that it pushes the DS-TWR method to its theoretical limit in maximizing the number of distance calculations.
3) We have implemented this protocol on Crazyflie 2.1 drones with onboard UWB wireless transceiver chips DW3000. Experiments with 25 drones show a 47.8% improvement over *SRv1* and over 300% improvement compared to standard UWB protocol, demonstrating the protocol's scalability and effectiveness in real-world swarm deployments.
4) To the best of our knowledge, we are among the first to provide a theoretically grounded many-to-many UWB ranging protocol. The analysis method in this paper sheds light on other UWB protocol designs in the community.

II. Preliminary and Motivation

A. Double-sided two-way ranging (DS-TWR) protocol

A standardized Ultra-Wideband (UWB) ranging protocol, defined in IEEE Standard 802.15.4z-2020 [10], is the double-sided two-way ranging (DS-TWR) protocol. Four types of messages are exchanged between the two sides, A and B, as illustrated in Fig. 1. The reply and round trip time durations for the two sides are defined as follows:

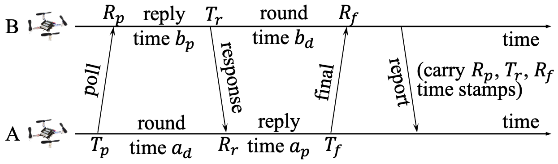


Fig. 1: The double-sided two-way ranging (DS-TWR) protocol.

$$a_d = R_r - T_p, b_p = T_r - R_p, b_d = R_f - T_r, a_p = T_f - R_r. \quad (1)$$

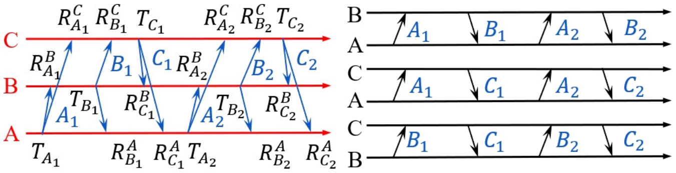
Let t_p represent the time of flight (ToF), which is the radio signal propagation time. Thus,

$$t_p = \frac{a_d b_d - a_p b_p}{a_d + b_d + a_p + b_p}. \quad (2)$$

B. The basic idea of swarm ranging version 1.0

SRv1 is a communication protocol designed to enable simultaneous ranging and data transmission in dynamic and dense networks. It leverages the broadcast nature of wireless transmission to extend DS-TWR protocols, thereby supporting robust and scalable devices or robots.

SRv1 utilizes only a single type of packet, known as the *ranging message*. A three-sides example is provided in Fig. 2, where messages are transmitted in sequence: A_1, B_1, C_1, A_2, B_2, and C_2. Given the broadcast nature of wireless communication, every message is received by the other two sides. As shown in Fig. 2(a), this process results in each message generating three timestamps. We can see that each pair has two rounds of message exchanges in (b). Therefore, each pair has sufficient timestamps to calculate the ToF.



(a) Each side periodically broadcasts *ranging* messages.

(b) Each pair has enough timestamps to calculate the ToF.

Fig. 2: A three-sides example illustrating the core idea of the *SRv1* protocol [8], [9].

By Eqs. (1) and (2), six timestamps are required to compute the distance, therefore each node maintains a *ranging table* for every neighbor. To compute the distance more frequently, the sliding-window technique is adopted to update the *ranging table* and perform computation. Fig. 3 (a) and (b) shows the full steps that the ranging tables are updated to correctly compute the distance between A and Y.

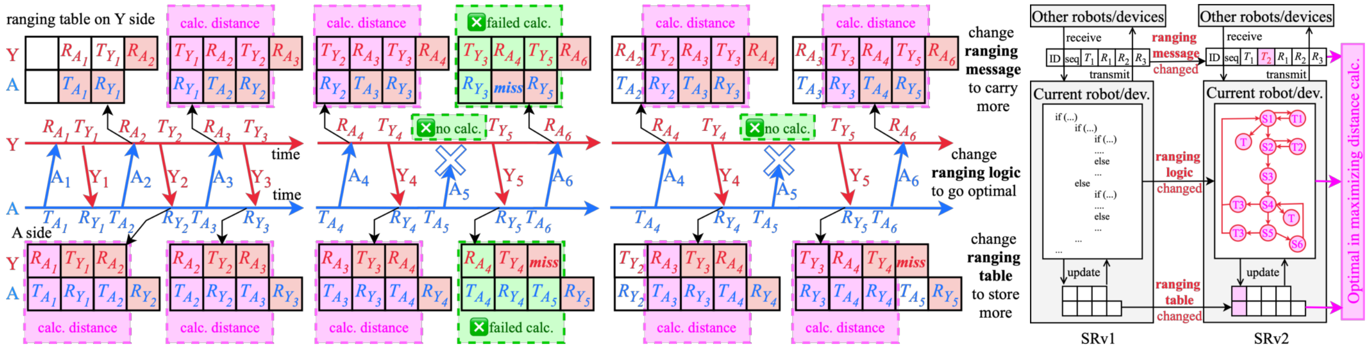
C. Motivations and Main Ideas

During our experiment, we identified two key drawbacks of *SRv1* that motivated us to design a new protocol that better supports dynamic and dense swarms.

1) On ranging messages and ranging table: We found that wireless packet loss is quite common in large swarm communication and ranging due to heavy channel load, yet the *SRv1* protocol handles it poorly. As shown in Fig. 3(b), when ranging message A_5 is lost, Y immediately loses one reception-triggered opportunity for distance computation. The following message Y_5 carries no timestamp R_{A_5}, so ranging fails upon receiving Y_5 at A due to an incomplete ranging table. Furthermore, A_6 carries T_{A_5} instead of T_{A_4}, so the ranging table is incomplete upon receiving A_6 at Y, preventing distance calculation. In conclusion, **three ranging opportunities are missed due to one single packet loss by *SRv1*.**

Our core idea is to redesign the *ranging table* and *ranging message*. For the *ranging table*, we expand it to store an additional pair of timestamps, as shown in Fig. 3(c). Upon receiving Y_5, although a critical timestamp is still missing, A can utilize the additional pair to compute the distance. This

[1] We use *packet* and *message* interchangeably in this paper.



(a) *SRv1* calc. dist. upon recv. (b) *SRv1*: 3 missed by 1 lose. (c) *SRv2*: 1 missed ranging by 1 packet lose. (d) *SRv1* versus *SRv2*.

Fig. 3: An illustration of how *SRv2* works. Compared to *SRv1*, (1) we modify the ranging table to store a pair of additional timestamps, which are critical for *compensatory ranging*; (2) we modify the ranging message to carry more timestamps, enabling additional distance calculations in packet loss condition; and (3) we replace the ranging handling logic with a state machine, which plays a pivotal role in proving that the protocol is optimal to maximize distance calculations.

ranging method has not been previously reported in related work, and we name it *compensatory ranging*. For the *ranging message*, we require it to carry additional timestamps, such that the ranging message A_6 carries not only T_{A_5} but also T_{A_4}. Upon receiving A_6, the critical timestamp in the ranging table is completed, and the distance can be computed. In conclusion, **only one distance ranging opportunity is missed for a single message loss in *SRv2***, which is inevitable.

For inconsistent ranging frequencies, where the two sides send ranging messages at different rates, a similar problem persists, as unsent packets are effectively equivalent to lost ones. Therefore, the method described above also applies.

2) On ranging protocol analysis: Since packet loss and inconsistent message transmission frequencies are common in dynamic and dense swarms, their combinations may lead to various cases. The *SRv1* protocol handled them on a case-by-case basis, resulting in deeply nested *if-else* conditionals, making its correctness hard to verify, let alone its optimality. Thus, we completely re-designed the ranging logic in *SRv2* using a state machine-based approach. The various cases caused by complex combinations are unified into three simple ranging events: `TX`, `RX`, and `RX_NO`. Then, the events trigger state transitions, which guide how to update the ranging table correctly. The introduction of state machine approach plays a pivotal role in proving that the protocol is optimal to maximize distance calculations. The differences are given in Fig. 3(d).

III. Design of Swarm Ranging 2.0 Protocol

A. Compensatory Ranging and Ranging Table Design

In dynamic and dense robotic or device swarms, packet loss and inconsistent ranging frequencies, where two sides send ranging messages at different rates, are common. We introduce the concept of ***Inconsistent Ranging Duration***, which refers to a sub-duration during which a robot/device A receives k packets ($k > 1$) from neighbor Y, while Y receives none from A, either due to packet loss or unsent packets. In Fig. 4(a), A receives k messages from Y during an *inconsistent ranging duration*. Fig. 3(b) shows another example with $k = 2$.

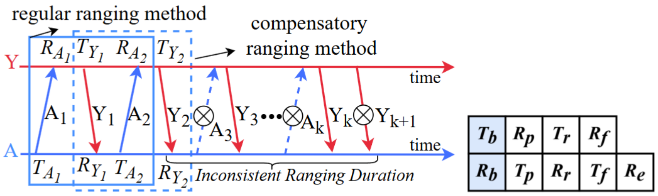
In any *inconsistent ranging duration*, at most one distance can be calculated by *SRv1* due to missing timestamps, using the *regular ranging method*. We define the ***regular ranging method*** as applying DS-TWR (Eq. (1)(2)) using six timestamps generated from three messages: the most recent `send-receive-send` sequence.

Focusing on the *inconsistent ranging duration*, we discovered a previously unexplored way to perform an additional distance calculation, and thus propose the *compensatory ranging method*. The ***compensatory ranging method*** is defined as applying DS-TWR (Eq. (1)(2)) for six timestamps generated from three messages: the most recent `receive-send-receive` sequence.

In Fig. 4(a), after A receives Y_2, the *regular ranging method* computes distance by applying Eq. (1)(2) to A_1, Y_1, and A_2. After receiving Y_k, the *compensatory ranging method* computes distance by applying Eq. (1)(2) to Y_1, A_2, and Y_{i-1}, where $k = 3, 4, \cdots, k+1$.

Although *compensatory ranging* increases the number of distances computed during an *inconsistent ranging duration*, it has side effects.

Lemma 1. *For an* inconsistent ranging duration *in which* k


regular ranging method
compensatory ranging method
Y
A
time
R_{A_1} T_{Y_1} R_{A_2} T_{Y_2}
A_1 Y_1 A_2 Y_2 A_3 Y_3 A_k Y_k Y_{k+1}
T_{A_1} R_{Y_1} T_{A_2} R_{Y_2}
Inconsistent Ranging Duration

T_b	R_p	T_r	R_f	
R_b	T_p	R_r	T_f	R_e

(a) In *SRv1*, by *regular ranging*, A computes distance by 6 timestamps from the latest `send--receive-send` messages. In *SRv2*, *compensatory ranging* computes with the latest `receive--send-receive` messages timestamps, enabling more rangings in *inconsistent ranging duration*.

(b) New *ranging table* with additions of T_b and R_b compared to *SRv1*'s ranging table.

Fig. 4: Structure of new *ranging table* and scenario of ***Inconsistent Ranging Duration***.

messages are received ($k > 1$)*, only the initial* compensatory ranging *calculation yields the most recent distance. Alternatively, any DS-TWR-based method benefits only from the first* compensatory ranging *calculation.*

Proof. See Appendix VIII-A. □

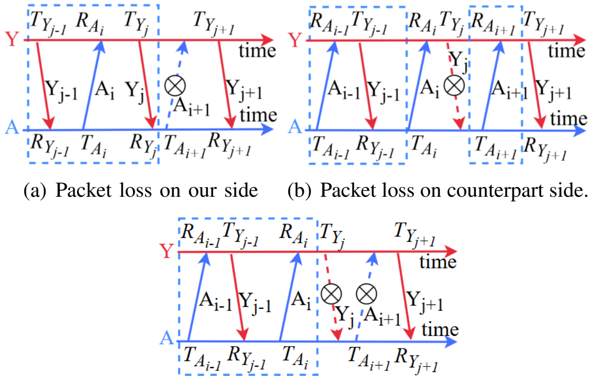
This lemma indicates that repeatedly performing compensatory ranging is inefficient. To address this, we redesign the *ranging table* to support both regular and compensatory ranging methods, as shown in Fig. 4(b). The redesign incorporates T_b and R_b to cache the T_r and R_r timestamps from the last successful ranging calculation using the regular ranging method and, as illustrated in Fig. 5, clears them after each compensatory ranging operation.

STEP	**New Ranging Table**				
A Receives Y$_2$ & Compute Distance	$T_b =$	$R_p = R_{A_1}$	$T_r = T_{Y_1}$	$R_f = R_{A_2}$	
Regular Ranging Method	$R_b =$	$R_p = T_{A_1}$	$R_r = R_{Y_1}$	$T_f = T_{A_2}$	$R_e = R_{Y_2}$
Rearrange the Table	$T_b = T_{Y_1}$	$R_p = R_{A_2}$	$T_r =$	$R_f =$	
	$R_b = R_{Y_1}$	$R_p = T_{A_2}$	$R_r = R_{Y_2}$	$T_f =$	$R_e =$
A Receives Y$_3$ & Compute Distance	$T_b = T_{Y_1}$	$R_p = R_{A_2}$	$T_r = T_{Y_2}$	$R_f =$	
Compensatory Ranging Method	$R_b = R_{Y_1}$	$R_p = T_{A_2}$	$R_r = R_{Y_2}$	$T_f =$	$R_e = R_{Y_3}$
Clear T_b and R_b , prevent	$T_b =$	$R_p = R_{A_2}$	$T_r = T_{Y_k}$	$R_f =$	
consecutive compensatory ranging	$R_b =$	$R_p = T_{A_2}$	$R_r = R_{Y_k}$	$T_f =$	$R_e = R_{Y_3}$
• • •					
A Receives Y$_{k+1}$	$T_b =$	$R_p = R_{A_2}$	$T_r = T_{Y_k}$	$R_f =$	
	$R_b =$	$R_p = T_{A_2}$	$R_r = R_{Y_k}$	$T_f =$	$R_e = R_{Y_{k+1}}$

Fig. 5: An example of A updates the new ranging table and computes the distance for the message arrival sequence in Fig. 4(a). The newly proposed *compensatory ranging method* increases distance computed in the *inconsistent ranging duration*. After each compensatory ranging, clearing T_b and R_b to prevent consecutive compensatory ranging.

B. Design of Ranging Message for Packet Loss

Building on the two ranging methods and the new *ranging table*, this section aims to redesign the *ranging message* and leverage both methods to better handle various packet loss scenarios. For clarity, we refer to the transmitting and receiving robots/devices as 'our side' and 'the counterpart's side.' As shown in Fig. 6, 'A side' represents 'our side'. This analysis will focus on the following three primary scenarios:



(a) Packet loss on our side (b) Packet loss on counterpart side.

(c) Simultaneous packet losses on both sides.

Fig. 6: Three primary packet loss scenarios.

1) Packet loss on our side: As illustrated in Fig. 6(a), the depicted scenario involves a single packet loss exclusively on our side. Clearly, this situation is equivalent to the ***Inconsistent Ranging Duration***. Thus, this issue can be addressed effectively using the newly introduced *ranging table* and *compensatory ranging* method, as detailed in Section III-A.

2) Packet loss on the counterpart's side: As shown in Fig. 6(b), the scenario depicted involves a single message loss from the counterpart, where our side fails to receive the Y_j message. We notice that if the message Y_{j+1} can carry the transmission timestamp of Y_{j-1}, then ranging can be performed using the 6 timestamps of ranging message A_{i-1}, Y_{j-1}, and A_{i+1}. More generally, including k last transmission timestamps in each message can tolerate up to $k-1$ consecutive packet losses on the counterpart's side.

3) Simultaneous Packet Loss on Our and the Counterpart's Sides: As illustrated in Fig. 6(c), this scenario shows a typical example where both Y_j and A_{i+1} are lost simultaneously. it can be noted that if Y_{j+1} carries the latest reception timestamp R_{A_i}, ranging can be performed using the six timestamps from messages A_{i-1}, Y_{j-1}, and A_i. Therefore, to enable ranging in such scenarios, each ranging message should always carry the latest reception timestamp of the neighbor's ranging message.

In conclusion, the above analysis suggests that the ranging message design must incorporate the following two rules.

Rule 1. *The* SRv2 *protocol incorporates multiple last transmission timestamps in the ranging message.*

Rule 2. *The* SRv2 *protocol always carries the latest reception timestamp for each neighbor in the ranging message.*

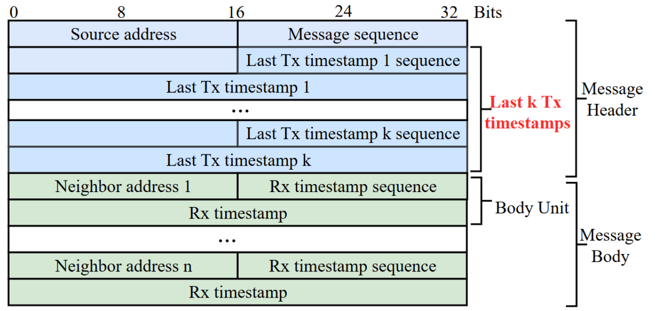


Fig. 7: Ranging message format in *SRv2*. To handle consecutive packet losses on the counterpart's side, the ranging message is modified to include the k most recent transmission (TX) timestamps. When $k = 1$, the message reduces to the original one in *SRv1*.

Accordingly, the newly designed *ranging message* is shown in Fig. 7, which ensures that even in cases of $k-1$ consecutive packet losses, the most recent available timestamps can still be effectively utilized for ranging. In practice, k is typically set to a small value (e.g., 3 to 5) for two reasons: (1) A small k is sufficient in low packet loss scenarios, as the probability of $k-1$ consecutive losses is significantly low; (2) A larger k may include outdated ranging information, resulting in outdated distance estimates. Therefore, carrying $k-1$ additional timestamps enables more robust ranging and only causes small overhead.

C. *State Machine Model for Efficient Swarm Ranging*

Given the ranging table and message designed in previous subsections, we now describe how to update the ranging table when ranging messages are received or transmitted. The ultimate goal is to calculate distances and update the ranging table in preparation for the next calculation.

When handling update operations, *SRv1* considers not only on which side the packet is lost, but also whether transmission frequencies are mismatched. The various cases caused by complex combinations result in deeply nested *if-else* conditionals, making its correctness hard to verify, let alone its optimality.

In contrast, *SRv2* adopts a state machine approach to systematically organize update operations. Since the possible ways in which the ranging table can be filled are finite, we model them as **states**. There are many such states, and transitions between them are triggered by **events**. Two basic events are obvious: ranging message transmitted and received. However, as illustrated in Fig. 4, a received ranging message may or may not carry a valid R_f. Therefore, we summarize three types of *events*: `TX`, `RX`, and `RX_NO`, where: `TX` represents a newly transmitted message; `RX` represents a newly received message carrying a valid R_f; `RX_NO` represents a newly received message without a valid R_f.

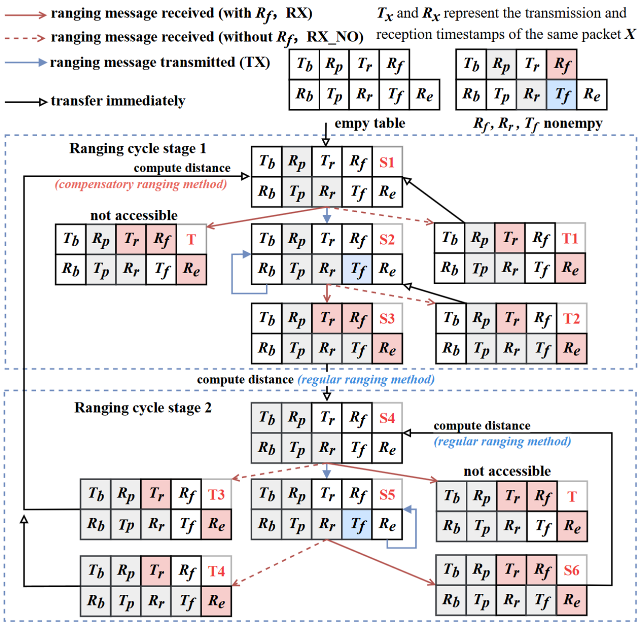

ranging message received (with R_f, RX)
ranging message received (without R_f, RX_NO)
ranging message transmitted (TX)
transfer immediately
T_X and R_X represent the transmission and reception timestamps of the same packet X
T_b R_p T_r R_f
R_b T_p R_r T_f R_e
empty table
R_f, R_r, T_f nonempty
Ranging cycle stage 1
compute distance
(compensatory ranging method)
S1
not accessible
T
S2
T1
S3
T2
compute distance (regular ranging method)
Ranging cycle stage 2
S4
compute distance
(regular ranging method)
not accessible
T3
S5
T
T4
S6

Fig. 8: The state machine for *SRv2* protocol. We divide the states into *Cycle Stage 1* and *Cycle Stage 2* based on whether *compensatory ranging* is possible. By any `TX` event, T_f is generated and table updated in blue; while by any `RX` event, R_e is generated and T_r and R_f are carried, and table updated in red. By Lemma 1, whenever a `RX_NO` event occurs in stage 2, the *compensatory ranging method* is invoked, and the state transits back to S1, to prevent further invalid compensatory ranging attempts.

A full set of states and event-driven state transitions is shown in Fig. 8. We divide the states into two groups based on whether (T_b, R_b) in the ranging table is ready, namely, *Cycle Stage 1* and *Cycle Stage 2*. *Compensatory ranging* does not occur in Cycle Stage 1 but becomes possible in Cycle Stage 2.

Cycle Stage 1 includes three primary states: S1, S2, and S3, as well as three temporary states: T, T1, and T2. State T is inaccessible since, without a transmission timestamp T_f, the receiving timestamp R_f cannot exist. In the temporary states T1 and T2, ranging is not possible because the required six complete timestamps are not yet available in the ranging table. After at least one `TX` event and one `RX` event, the state transitions from S3 to S4, thereby entering Cycle Stage 2.

In Cycle Stage 2, under ideal conditions, a ranging calculation can be performed using the regular ranging method following each `TX` and `RX` event. State transitions occur exclusively among states S4, S5, and S6.

According to Lemma 1, only the first compensatory ranging within an *inconsistent ranging duration* is valid. Therefore, when a `RX_NO` event occurs, the *compensatory ranging method* is invoked to compute distance, and (T_b, R_b) is cleared to prevent further invalid compensatory ranging attempts. This causes the state to transition back to S1.

D. SRv2 *Protocol Optimality Analysis*

The goal of this subsection is to analyze the optimality of the *SRv2* Protocol.

Theorem 1. *The* SRv2 *protocol is the optimal protocol that maximizes the count of distance calculations.*

Proof. We prove this by contradiction. Suppose there exists another protocol, named *OPT*, that generates more distance calculations than *SRv2*. We aim to find contradictions. Since the *OPT* protocol may adopt a totally different message format, carrying a totally different set of information.

To make the two protocols comparable, we further restrict that they work on the same set of messages. Assume A is ranging with Y, one of its neighbors. Let the transmitted message set by A be $\mathcal{M}_A = \{A_1, A_2, \cdots\}$, and the received message set by Y be $\mathcal{M}'_A \subseteq \mathcal{M}_A$. Similarly, we define the message set from Y as $\mathcal{M}_Y = \{Y_1, Y_2, \cdots\}$, and set received by A as $\mathcal{M}'_Y \subseteq \mathcal{M}_Y$. We further assume that $\mathcal{M}_A(A_i)$ and $\mathcal{M}'_A(A_i)$ are the transmit and reception timestamps of message A_i, respectively. Both *SRv2* and *OPT* work on message sets $\mathcal{M}_A$ and $\mathcal{M}'_Y$, although the message content may be different.

We define the ranging count by *SRv2* as $ranging(SRv2, \mathcal{M}_A, \mathcal{M}'_Y)$, or $ranging(SRv2)$ for short; the ranging count by *OPT* as $ranging(OPT, \mathcal{M}_A, \mathcal{M}'_Y)$, or $ranging(OPT)$.

Now we compute $ranging(SRv2)$. Since the *SRv2* protocol works according to the state machine in Fig. 8, we can easily see that only four state transitions involve ranging, *i.e.,* $S3 \rightarrow S4, S6 \rightarrow S4, T3 \rightarrow S1$, and $T4 \rightarrow S1$. We further realize that some events at these states can trigger ranging. Accordingly, we define $c(S, E)$ as the total count of event E at state S during the entire process.

Then, $ranging(SRv2) = c(S2, RX) + c(S4, RX\_NO) + c(S5, RX) + c(S5, RX\_NO)$. Moreover, we have the total number of received messages $|\mathcal{M}'_Y|$ be either RX_NO or RX event, so $|\mathcal{M}'_Y| = c(S1, RX\_NO) + c(S2, RX) + c(S2, RX\_NO) + c(S4, RX\_NO) + c(S5, RX) + c(S5, RX\_NO)$.

Since any protocol, including *OPT*, can only update the distance upon the reception of a message, so $ranging(OPT) \leq |\mathcal{M}'_Y|$. Now we can see the ranging count difference between the two protocols, $ranging(OPT) - ranging(SRv2) \leq c(S1, RX\_NO) + c(S2, RX\_NO)$.

Next, we focus on two state transitions $S1 \xleftrightarrow{\text{RX\_NO}} T1$ and $S2 \xleftrightarrow{\text{RX\_NO}} T2$, whose occurrence counts are $c(S1, RX\_NO)$ and $c(S2, RX\_NO)$ respectively. From the state machine in Fig. 8, we can see that once the state transitions from $S1$ to $T1$ (or from $S2$ to $T2$), it immediately returns to the original state. This indicates that the two state transitions may occur continuously multiple times. Related transitions are given as below.

$$\begin{array}{lcl} S4 \xrightarrow{\text{RX\_NO}} T3 \rightarrow & \multirow{2}{*}{$S1$} & \xleftrightarrow{\text{RX\_NO}} T1 \\ S5 \xrightarrow{\text{RX\_NO}} T4 \rightarrow & & \xrightarrow{\text{TX}} S2 \xleftrightarrow{\text{RX\_NO}} T2 \end{array}$$

Without loss of generality, suppose the first transition occurs continously for a times, and the second for b times, with $k = a + b$. Note that these k state transitions are driven by a set of `RX_NO` events. Let $Y_i \in \mathcal{M}'_Y$ be the initial `RX_NO` event, and $Y_j \in \mathcal{M}'_Y$ be the final `RX_NO` event. Let $t_i = \mathcal{M}'_Y(Y_i)$ and $t_j = \mathcal{M}'_Y(Y_j)$. Then the period (t_i, t_j) is the *inconsistent ranging duration* we focused on.

The core idea of our optimality proof is to demonstrate that in *inconsistent ranging duration* (t_i, t_j), although $SRv2$ computes no distance, *OPT* also computes none. The key connection between $SRv2$ and OPT is that they share the same set of messages exchanged, *i.e.,* $\mathcal{M}_A$, $\mathcal{M}'_A$, $\mathcal{M}_Y$, and $\mathcal{M}'_Y$, although OPT may adopt a completely different message format, carrying an entirely different set of information. We have the following lemma.

Lemma 2. *For any A_k that satisfies $A_k \in \mathcal{M}_A$ and $t_i < t_k < t_j$, where $t_k = \mathcal{M}_A(A_k)$, then $A_k \notin \mathcal{M}'_A$.*

Proof. We prove by contradiction. If $A_k \in \mathcal{M}'_A$, then by Rule 2, the *SRv2* always carries the latest reception timestamp for each neighbor in the transmitted message, This implies that an RX event rather than an RX_NO event shall occur, which is a contradiction. □

Therefore, according to Lemma 2 in the duration (t_i, t_j), if there is a message sent from A it must have been lost. For such message sets $\mathcal{M}_A$ and $\mathcal{M}'_Y$, $ranging(OPT) \geq ranging(SRv2)$, then distance must be computed at S1 or S2 receiving an RX_NO event, in other words, continuously compensatory ranging must have occurred. This is a contradiction to Lemma 1. Therefore, our *SRv2* protocol is optimal. □

IV. Implementation

The proposed swarm ranging 2.0 protocol has been implemented using Crazyflie 2.1 drones, which are micro drones powered by STM32 microcontrollers, equipped with 192KB of memory, and onboard UWB (Ultra-Wideband) wireless transceiver chips DW3000, as illustrated in Fig. 9.

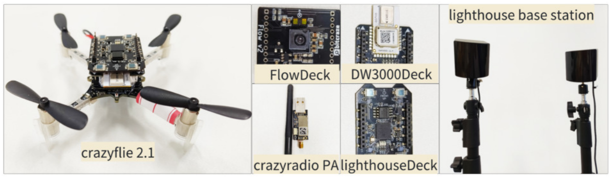


Fig. 9: *SRv2* protocol experimental platform based on Crazyflie, equipped with FlowDeck for height estimation, DW3000 deck for UWB communication, Lighthouse for ground truth, and Crazyradio PA for data collection.

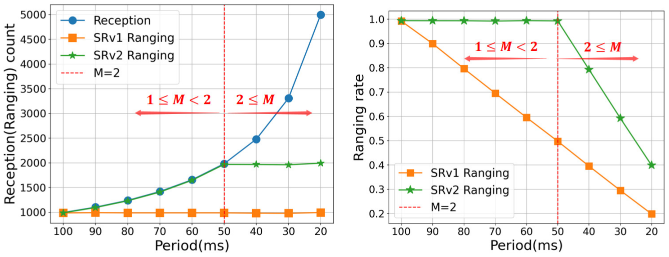
The communication setup uses a data rate of 6.8 Mbps with a 128-bit preamble. Drone data are transmitted to a laptop via Crazyradio PA (2.4 GHz). Each drone is equipped with a Flow Deck for automatic flight control and height estimation. For experiments requiring ground truth, the HTC Lighthouse system with millimeter-level accuracy is used, with a Lighthouse deck mounted on each drone for position acquisition.

V. Experiment

This section evaluates the swarm ranging protocol 2.0 in multiple aspects through experiments.

A. Performance for Inconsistent Ranging Duration

To evaluate the performance of *SRv2* under ***Inconsistent Ranging Duration***, we conducted an experiment with two drones, A(fixed 100 ms transmission period) and B(varying from 100 ms to 20 ms). Experimental data were collected from A for 100 seconds, as shown in Fig. 10. Here, we defined ***Inconsistent Frequency Degree*** M as $M = \frac{P_l}{P_s}$, where P_l and P_s denote the periods of the long- and short-period messages, respectively. Clearly, $M \geq 1$.



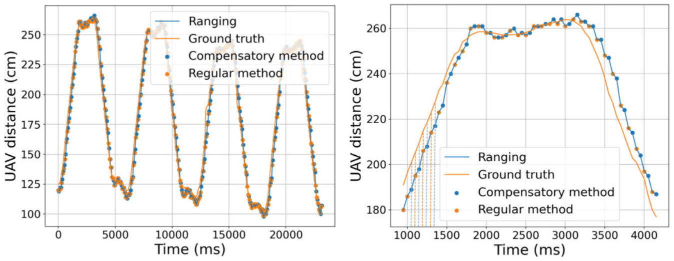
(a) Comparison of ranging count (b) Comparison of ranging rate

Fig. 10: Comparison of ranging count and rate(ranging count / reception count) under different period mismatch degree in *SRv1* and *SRv2* protocols

The results show that, in *SRv1*, as the period mismatch degree(M) increases, the count of received packets gradually increases, while the count of ranging remains nearly constant. In contrast, the *SRv2* protocol shows considerable improvement. When $M \leq 2$, increasing drone B's transmission frequency leads to a higher ranging count, with the ranging rate staying close to 1. However, when $M > 2$, the ranging count no longer increases, causing a decline in the ranging rate. This is because, according to Lemma 1, multiple consecutive compensatory ranging events occur, of which only the first is valid.

In summary, when $M \leq 2$, high-frequency ranging messages can be fully utilized, maximizing the number of ranging and reaching the theoretical limit of DS-TWR method. When $M > 2$, further increasing transmission frequency no longer improves ranging count.

B. Accuracy of Ranging in Dynamic Scenarios



(a) Four cycles of move-away and move-close motion

(b) Zoomed-in view of a segment from (a).

Fig. 11: Accuracy of compensatory and regular ranging during flight. Drone A hovers steadily, while drone B flies at a speed of 1 m/s, repeatedly moving close and away over multiple cycles. Transmission period was 100 ms for A and 50 ms for B, resulting in one compensatory ranging performed between every two regular ranging. Ground truth is added for comparison.

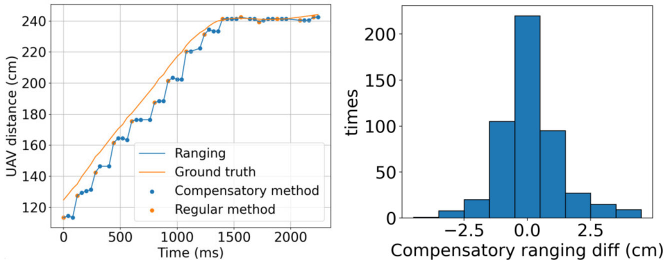
To evaluate the accuracy of regular ranging and compensatory ranging in dynamic scenarios, we conducted two experiments with drones A and B.

TABLE I. Comparison of the mean and standard variance of the difference between ground truth and ranging data for regular and compensatory methods

	Regular Method		Compensatory Method	
	Mean(cm)	Standard Deviation	Mean(cm)	Standard Deviation
Move away	8.34	2.57	9.23	3.15
Move close	-6.72	3.07	-9.25	3.41

In the first experiments, A and B have transmission periods of 100 ms and 50 ms, respectively. Data were collected from A, as shown in Fig. 11(a), (b) and Table I. Fig. 11(a) compares the ranging data with the ground truth from four cycles, while Fig. 11(b) shows a zoomed-in view of a portion of the data from (a). According to Fig. 11(b), when drone B moves away from drone A (1000–1700 ms), the inter-drone distance increases, but the measured distance is consistently smaller than the ground truth due to protocol-induced latency. Conversely, when drone B moves close to drone A (3300–4000 ms), the distance decreases, while the measured distance exceeds the ground truth.

Table I presents the comparison of the ranging accuracy between the two ranging methods and the ground truth. Although the accuracy of the compensatory ranging is slightly lower than that of the regular ranging, it nonetheless remains effective in updating the inter-drone distance, as illustrated in Fig. 11(b). This demonstrates that the proposed protocol can increase the overall ranging frequency by combining regular and compensatory ranging methods.



(a) Ranging results of the two methods compared with ground truth

(b) Distribution of differences between the second/third compensatory ranging results relative to the first

Fig. 12: Validation of consecutive compensatory ranging: Drone A hovers steadily, while drone B flies at a speed of 1 m/s, moving close and away relative to A. Transmission period was 160 ms for A and 40 ms for B, resulting in three consecutive compensatory ranging.

The data from the second experiment are shown in Fig. 12. In Fig. 12(a), ranging data forms a stepped curve, with three consecutive blue points(consecutive compensatory ranging) between two orange points(regular ranging) remaining nearly the same. Fig. 12(b) shows the difference distribution of the second and third compensatory ranging value relative to the first. It can be observed that most of the differences are distributed around zero, indicating that only the first compensatory ranging is valid, thereby validating Lemma 1.

C. Comparison with SRv1 and IEEE 802.15.4z in dense swarms

To evaluate the *SRv2's* ranging performance in dense swarms, we conducted a static comparative study involving *SRv2*, *SRv1* and the IEEE 802.15.4z standard. In this experiment, the ranging periods were randomized between 40 and 80 ms and each message carries 4 last transmission timestamps.

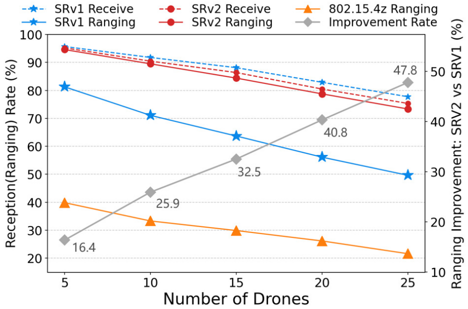


Fig. 13: Comparison of receiving and ranging rate among *SRv1*, *SRv2* and Extension DS-TWR in IEEE Standard 802.15.4z-2020. **As the number of drones increases from 5 to 25**, *SRv2* reception rate(red dashed line) remains slightly below *SRv1* reception rate(blue dashed line). However, *SRv2* ranging rate(red solid line) is significantly higher than *SRv1* ranging rate(blue solid line). The gray line illustrates the ranging improvement ratio of *SRv2* over *SRv1*.

The results in Fig. 13 show that *SRv2* exhibits a slightly lower reception rate than *SRv1* as the number of drones increases. This is primarily due to the additional processing overhead introduced by the extra timestamp information carried in *SRv2* ranging message. However, its ranging rate improves significantly, with the gain becoming more pronounced as the swarm size grows. Specifically, **with 25 drones**, *SRv2* achieves a 47.8% increase in ranging rate over *SRv1*, and nearly a threefold improvement over standard protocol. This demonstrates the robustness and scalability of *SRv2* in dense swarm environments.

D. Demonstration experiment

We conducted an eight-drone swarm formation experiment to assess in-flight ranging performance. Each drone's period was randomized between 40 and 80ms. Initially, Drone 0 was centrally positioned, with other drones arranged in a 2x2m square around it, as shown in Fig. 14. During flight, while Drone 0 hovered, other drones executed a coordinated rotational formation around it. Ranging data were collected from Drone 1 relative to three others, as depicted in Fig. 15. Although occasional deviations arise due to systematic errors from UWB antenna obstructions, ranging values generally remain close to ground truth. Fig. 15(b) shows the ranging error distribution, revealing a Mean Absolute Error (MAE) of 7.19 cm (the ranging accuracy specified in the official Qorvo manual [11] is 10 cm).

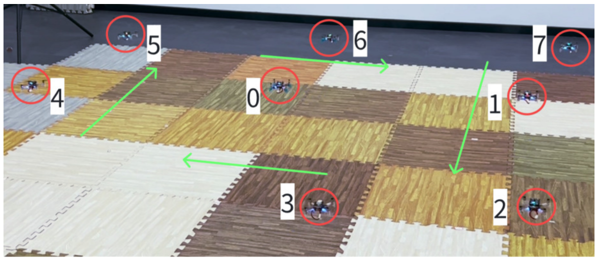
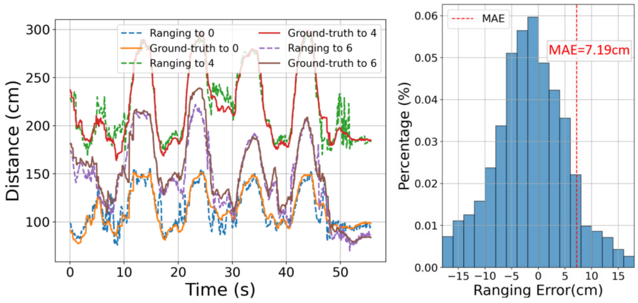


Fig. 14: Top view of 8 drones flying in formation in a $2\,\mathrm{m} \times 2\,\mathrm{m}$ square area.



(a) Ranging vs. Ground Truth over Time: drones random flight for the first 10s, formation flight from 10s to 50s, and landing afterward

(b) Frequency distribution of ranging error relative to ground truth, along with the mean absolute error (MAE).

Fig. 15: Ranging vs. Ground Truth: Distance measurements from drone 1 to three other drones during an 8-drone formation flight.

VI. Related Work

UWB is widely used for indoor localizations [12], [13], [14]. Aditya *et al.* [12] develop XRLoc, providing an accuracy of a few centimeters with a single anchor in many real scenarios. Valerio *et al.* [13] propose a UWB-based relative localization system where ranging data are collected via BLE connections to localize four nodes. Miramá *et al.* [14] use an empirical approach based on ML models applied to indoor pedestrian localization based on 6.5 GHz UWB devices.

Error correction in UWB ranging has consistently been a focal point of research interest [15], [16], [17]. Liu *et al.* [16] presents two machine learning approaches from UWB sensors deployed on a highway bridge and outperformed the state-of-the-art approaches in terms of measurement accuracy and output frequency. Margiani *et al.* [17] use a compact, low-power solution integrating a novel commercial module with Phase Difference of Arrival estimation as integrated feature. Ma *et al.* [18] proposes a novel system that achieves sub-millimeter-level ranging accuracy.

There are also a few works focusing on UWB ranging for large numbers [19], [20] or for high mobility [21]. Corbalan *et al.* [19] focus to locate countless tag by TDOA. Moron *et al.* [20] proposes the implementation of a UWB role allocation algorithm within smart contracts on a blockchain to enhance scalability. Their work is dedicated for the anchor-tag model, not applicable for swarm scenario.

VII. Conclusion

This paper proposes an optimal UWB swarm ranging protocol for dynamic and dense robotic or device swarms, Swarm Ranging 2.0, which is theoretically proven to push the DS-TWR method to its performance limit. Firstly, it identified the limitations of Swarm Ranging 1.0 protocol within dynamic and dense swarm. To address them, a novel compensatory ranging method was introduced, along with a redesigned ranging message and table structure. A state machine model was subsequently employed for protocol design and validation, enabling robust handling of complex scenarios such as various combinations of packet loss and inconsistent message transmission frequencies. The proposed protocol was theoretically proven to achieve optimal ranging performance that pushes the DS-TWR method to its theoretical limit. Finally, the protocol was implemented on Crazyflie 2.1 drones equipped with DW3000 UWB transceiver chips, and real-world experiments validated its theoretical optimality and practical effectiveness.

VIII. Appendix

A. Proof of Lemma 1

Without loss of generality, we analyze the validity of the ranging process under the condition of receiving $k+1$ consecutive messages, where k compensatory ranging are performed. As illustrated in Fig. 16, consider Y moving towards A with a relative velocity v. Let P represent the ranging period for Y, during which A moves a distance vP. Assume that A first receives message Y_{i-1}, then transmits A_i, and subsequently receives $k+1$ consecutive messages from Y, corresponding to

$$t_{k+3}^{computed} = \frac{a_{d_k} \times b_{d_k} - a_{p_k} \times b_{p_k}}{a_{d_k} + b_{d_k} + a_{p_k} + b_{p_k}} \tag{3a}$$

$$= t_2 + \frac{t_2 t_\triangle(\beta - \alpha) + (\beta a_p - \alpha b_p)t_\triangle - \alpha\beta t_\triangle^2 + (k-1)t_\triangle(-t_2 + \beta P - b_p - \beta t_\triangle)}{4t_2 + 2a_p + 2b_p + t_\triangle(\beta - \alpha) + (k-1)(2P - t_\triangle)} \tag{3b}$$

$$= t_2 + \frac{t_2 t_\triangle(\beta - \alpha) + (\beta a_p - \alpha b_p)t_\triangle - \alpha\beta t_\triangle^2 + (k-1)t_\triangle t_2}{4t_2 + 2a_p + 2b_p + t_\triangle(\beta - \alpha) + (k-1)(2P - t_\triangle)} \approx t_2 \tag{3c}$$

distances $d_1, d_2, \ldots, d_{k+3}$. The time intervals $t_1, t_2, \ldots, t_{k+3}$ represent the wireless signal travel time. Define $t_\triangle$ as the time it takes for the wireless signal to travel the distance vP, given by $t_\triangle = \frac{vP}{c}$, where c is the speed of light. The reception timestamp of A_i divides the transmission period into a $\beta : \alpha$ ratio, where $\alpha + \beta = 1$. The symbols used and their interrelationships are further detailed in Fig. 16.

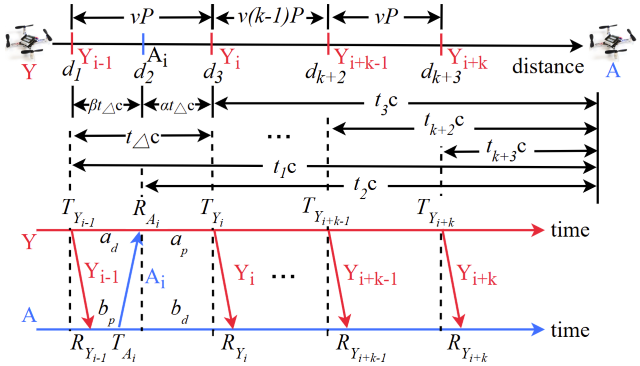


Fig. 16: Ranging accuracy and validation in dynamic scenarios with consecutive message receptions and no intervening transmissions.

Therefore, the equations $t_1 = t_2 + \beta t_\triangle$ and $t_{k+3} = t_2 - (\alpha + k)t_\triangle$ are established. The terms $a_d = R_{A_i} - T_{Y_{i-1}}$, $b_d = R_{Y_i} - T_{A_i}$, $a_p = T_{Y_i} - R_{A_i}$, and $b_p = T_{A_i} - R_{Y_{i-1}}$ are defined to clarify the variables used in calculations. Upon receiving the message Y_{i+k}, where the accurate ToF should correspond to t_{k+3}, the estimated ToF $t_{k+3}^{computed}$ is calculated using the six timestamps from the messages Y_{i-1}, A_i, and Y_{i+k-1}. Therefore, we define

$$\begin{aligned} b_{p_k} &= T_{A_i} - R_{Y_{i-1}} = b_p, \\ a_{d_k} &= R_{A_i} - T_{Y_{i-1}} = a_d = b_p + 2t_2 - \beta t_\triangle, \\ a_{p_k} &= T_{Y_{i+k-1}} - R_{A_i} = a_p + (k-1)P, \\ b_{d_k} &= R_{Y_{i+k-1}} - T_{A_i} = a_{p_k} + 2t_2 - (\alpha + k - 1)t_\triangle. \end{aligned}$$

Further derivation employing the DS-TWR calculation method, as illustrated by Equation 2, yields Equation 3. It is clear that irrespective of variations in k, the calculated time is always approximately equal to t_2, and thus the calculated distance approximates d_2.

B. An Example of Ranging Using the Final State Machine

Fig. 17 presents an example of ranging message transmission and reception between A and Y. Fig. 18 uses the ranging state machine to execute the example, illustrating the complete process of ranging and state transitions.

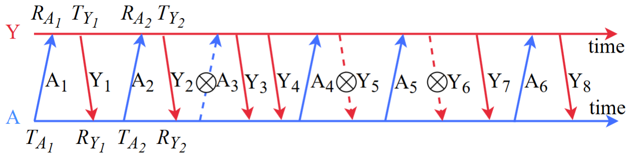


Fig. 17: An example of simulated packet transmission and reception between nodes A and Y, in which packet A3 from A and Y5 and Y6 from Y are lost. For readability, only partial timestamps are shown.

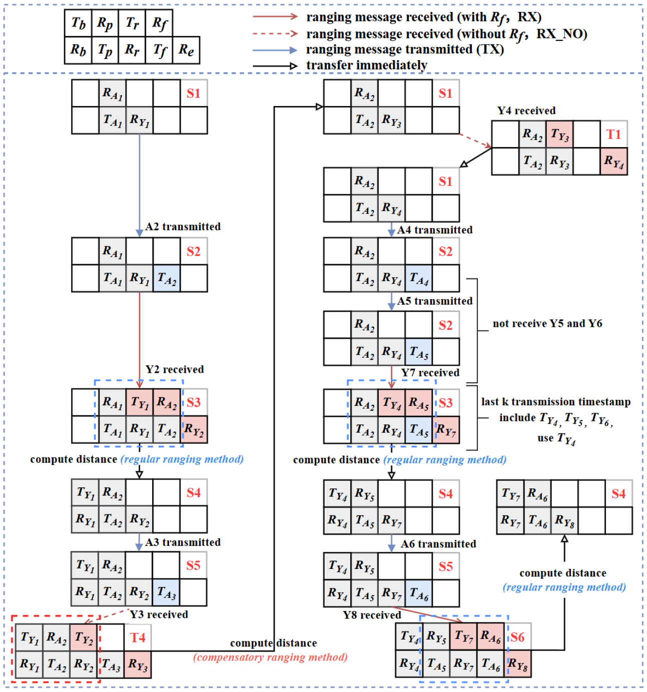

Tb Rp Tr Rf
Rb Tp Rr Tf Re
ranging message received (with Rf, RX)
ranging message received (without Rf, RX_NO)
ranging message transmitted (TX)
transfer immediately
S1
S2
S3
S4
S5
T4
S6
T1
A2 transmitted
Y2 received
compute distance (regular ranging method)
A3 transmitted
Y3 received
compute distance (compensatory ranging method)
Y4 received
A4 transmitted
A5 transmitted
not receive Y5 and Y6
Y7 received
last k transmission timestamp include T_{Y_4}, T_{Y_5}, T_{Y_6}, use T_{Y_4}
A6 transmitted
Y8 received
compute distance (regular ranging method)

Fig. 18: For the example in Fig. 17, the final state machine illustrates node A's complete ranging process. The blue dashed box indicates regular ranging, while the red dashed box indicates compensatory ranging. Upon receiving packet Y4, no ranging is performed because a compensatory ranging has already been completed previously and T_{Y_1}, R_{Y_1} is cleared (according to Lemma 1, only the first compensatory ranging in a sequence of consecutive compensatory rangings is valid). Ranging cannot be performed in state T1; in all other states, ranging can be completed upon receiving a packet.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China Grants 62232004, 62472090, 62132009, 62572115, Jiangsu Provincial Key R&D Program No. BE2022065-4, the National Natural Science Foundation of Jiangsu Grants BK20242026, the Jiangsu Provincial Key Laboratory of Network and Information Security Grant BM2003201, the Collaborative Innovation Center of Novel Software Technology.

REFERENCES

[1] S. Xu, X. Hu, R. Yang, C. Zang, L. Li, Y. Xiao, W. Liu, B. Tian, W. Pang, R. Bo *et al.*, "Transforming machines capable of continuous 3d shape morphing and locking," *Nature Machine Intelligence*, pp. 1–13, 2025.

[2] W. Shen, J. Peng, R. Ma, J. Wu, J. Li, Z. Liu, J. Leng, X. Yan, and M. Qi, "Sunlight-powered sustained flight of an ultralight micro aerial vehicle," *Nature*, vol. 615, no. 8021, pp. 1476–4687, 2024. [Online]. Available: https://doi.org/10.1038/s41586-024-07609-4

[3] "Crazyflie 2.1 - Bitcraze Store," https://www.bitcraze.io/2019/02/the-crazyflie-2-1-is-here/, accessed Jul. 30, 2020.

[4] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.

[5] Q. Wang, Q. Wang, Z. Ning, K. F. Chan, J. Jiang, Y. Wang, L. Su, S. Jiang, B. Wang, B. Y. M. Ip *et al.*, "Tracking and navigation of a microswarm under laser speckle contrast imaging for targeted delivery," *Science Robotics*, vol. 9, no. 87, p. eadh1978, 2024.

[6] Y. Liu, H. Chen, Q. Zou, X. Du, Y. Wang, and J. Yu, "Automatic navigation of microswarms for dynamic obstacle avoidance," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2770–2785, 2023.

[7] V. Niculescu, T. Polonelli, M. Magno, and L. Benini, "Ultra-lightweight collaborative mapping for robot swarms," *arXiv preprint arXiv:2407.03136*, 2024.

[8] F. Shan, J. Zeng, Z. Li, J. Luo, and W. Wu, "Ultra-wideband swarm ranging," in *Proc. INFOCOM*. IEEE, 2021, pp. 1–10.

[9] F. Shan, H. Huo, J. Zeng, Z. Li, W. Wu, and J. Luo, "Ultra-wideband swarm ranging protocol for dynamic and dense networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 6, pp. 2834–2848, 2022.

[10] "IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques," *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, pp. 1–174, 2020.

[11] Qorvo, Inc., *DW3000 Datasheet*, Online, 2024, available at: https://www.qorvo.com/products/d/da008334.

[12] A. Arun, S. Saruwatari, S. Shah, and D. Bharadia, "XRLoc: Accurate UWB localization to realize XR deployments," in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, 2023, pp. 459–473.

[13] V. Brunacci, A. De Angelis, G. Costante, and P. Carbone, "Development and analysis of a UWB relative localization system," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[14] V. Miramá, A. Bahillo, V. Quintero, and L. E. Díez, "NLOS detection generated by body shadowing in a 6.5 GHz UWB localization system using machine learning," *IEEE Sensors Journal*, 2023.

[15] Z. Qin, Z. Meng, Z. Li, N. Gao, Z. Zhang, Q. Meng, and D. Zhen, "Compensating the NLoS occlusion errors of UWB for pedestrian localization with MIMU," *IEEE Sensors Journal*, vol. 23, no. 11, pp. 12 146–12 158, 2023.

[16] Y. Liu and Y. Bao, "Real-time remote measurement of distance using UWB sensors," *Automation in Construction*, vol. 150, p. 104849, 2023.

[17] T. Margiani, S. Cortesi, M. Keller, C. Vogt, T. Polonelli, and M. Magno, "Angle of arrival and centimeter distance estimation on a smart UWB sensor node," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.

[18] J. Ma, F. Zhang, B. Jin, C. Su, S. Li, Z. Wang, and J. Ni, "Push the Limit of Highly Accurate Ranging on Commercial UWB Devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 2, pp. 1–27, 2024.

[19] L. Santoro, M. Nardello, D. Brunelli, and D. Fontanelli, "UWB-based indoor positioning system with infinite scalability," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.

[20] P. T. Morón, S. Salimi, J. P. Queralta, and T. Westerlund, "UWB role allocation with distributed ledger technologies for scalable relative localization in multi-robot systems," in *2022 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. IEEE, 2022, pp. 1–8.

[21] S.-H. Bach, P.-B. Khoi, and S.-Y. Yi, "Global UWB System: A High-Accuracy Mobile Robot Localization System With Tightly Coupled Integration," *IEEE Internet of Things Journal*, 2024.

[22] A. Li, E. Bodanese, S. Poslad, T. Hou, K. Wu, and F. Luo, "A trajectory-based gesture recognition in smart homes based on the ultrawideband communication system," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 861–22 873, 2022.

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

[24] Z. Zhang, F. Liu, and T. Zhang, "Fundamental limits on integrated sensing and communications frameworks: An IR-UWB case," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 2729–2734.