Ultra-Wideband Swarm Ranging Protocol for Dynamic and Dense Networks

Feng Shan, *Member, IEEE*, Haodong Huo, *Graduate Student Member, IEEE*, Jiaxin Zeng, Zengbao Li, Weiwei Wu, *Member, IEEE*, and Junzhou Luo, *Member, IEEE, ACM*

***Abstract*—Nowadays, not only wearable and portable devices but also aerial and ground robots can be made smaller, lighter, cheaper, and thus as large as hundreds of them may form a swarm to participate in a complicated cooperative application, such as searching, rescuing, mapping, and war-battling. Devices and robots in such a swarm have three important features, namely, large number, high mobility and short distance, hence they form a dynamic and dense wireless network. Successful swarm cooperative applications require low latency communications and real-time localization. This paper proposes to use ultra-wideband (UWB) radio technology to implement both functionalities, because UWB is very time-sensitive that an accurate distance can be calculated using the transmission and reception timestamps of data messages. A UWB swarm ranging protocol is designed to achieve simultaneously wireless data communication and swarm ranging that allows a device/robot to compute the distances to all the peer neighbors at the same time. This protocol is designed for dynamic and dense networks, meanwhile it can also be used in various wireless networks and implemented on various types of devices/robots including low-end ones. In our experiment, this protocol is implemented on Crazyflies, STM32 microcontroller powered micro drones, with onboard UWB wireless transceiver chips DW1000. Extensive real-world experiments are conducted to verify the proposed protocol on various performance aspects, with a total of 9 Crazyflie drones in a compact area. The implemented swarm ranging protocol is open-sourced at https://github.com/SEU-NetSI/crazyflie-firmware**

***Index Terms*—Ultra-wideband, swarm, ranging, protocol design, networking, two way ranging, Crazyflie.**

Manuscript received 24 July 2021; revised 4 May 2022; accepted 11 June 2022; approved by IEEE/ACM Transactions on Networking Editor R. Lo Cigno. Date of publication 29 June 2022; date of current version 20 December 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101200; in part by the National Natural Science Foundation of China under Grant 62072101, Grant 61972086, Grant 62072102, Grant 62172091, and Grant 62132009; in part by the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201; in part by the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China under Grant 93K-9; and in part by the Zhejiang Laboratory under Grant 2019NB0AB05. The preliminary version of this paper was presented in part at the 2021 IEEE International Conference on Computer Communications (IEEE INFOCOM 2021) [DOI: 10.1109/INFOCOM42981.2021.9488717]. *(Corresponding authors: Feng Shan; Junzhou Luo.)*

Feng Shan, Weiwei Wu, and Junzhou Luo are with the School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: shanfeng@seu.edu.cn; weiweiwu@seu.edu.cn; jluo@seu.edu.cn).

Haodong Huo and Zengbao Li are with the School of Cyber Science and Engineering, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: huohaodong@seu.edu.cn; lizengbao@seu.edu.cn).

Jiaxin Zeng is with the Technology Development Center, China UnionPay Company Ltd., Shanghai 200135, China (e-mail: zengjiaxin@unionpay.com).

Digital Object Identifier 10.1109/TNET.2022.3186071

I. Introduction

With the rapid development of the electronic manufacturing industry, more and more wearable and portable devices, aerial and ground robots are becoming commercially available. For example, smartphones and smartwatches from various vendors are being constantly released and upgraded, Bitcraze released the micro drone Crazyflie 2.1 in Feb. 2019 [1], DJI released the RoboMaster EP in March 2020 [2], Boston Dynamics launched commercial sales of SPOT legged robot in June 2020 [3].

Nowadays not only wearable and portable devices but also aerial and ground robots can be made smaller in size, lighter in weight, and cheaper in price, so they become more and more widely used in many applications. It is now possible to utilize as large as hundreds of them to form a swarm to participate in a complicated cooperative application. The following are few examples, a team of indoor drones searching for a given target, a swarm of small robots exploring and mapping an unknown indoor environment, an army of legged robot dogs battling in a deep forest. Compared with a large single robot with full functionality, a swarm of small devices and robots has the advantages of higher fault tolerance, more flexibility in deployment size and numbers, faster deployment speed. As a result, the research on device and robot swarm is a current trend, and publications are emerging in top journals such as *Nature* and *Science* [4]–[7].

We notice that the devices and robots in a swarm have three important features, namely, *large number*, *high mobility* and *short distance*. First, according to the task complexity, hundreds of devices and robots may be deployed to collaborate. Second, wearable and portable devices carried by human, micro drones, wheeled and legged robots, all can quickly respond to a request and make movements. Third, because of small sizes, they can cooperate within short distances in order to complete complicated tasks. Therefore, devices and robots in swarms from the future applications are expected to be *dynamic* and *dense*, hence they form a dynamic and dense wireless network for collaboration and communication.

Successful swarm cooperative applications require low latency communications and real-time localization. Especially, when there are no outside supportive infrastructures, it is critical to perform ad hoc networking and relative positioning among in-swarm devices and robots. Therefore, an efficient inter-device/robot communication protocol that supports

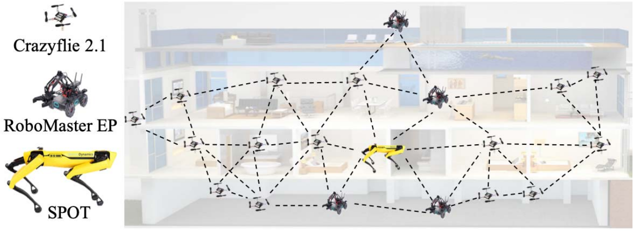


Fig. 1. Device and robot swarms are dynamic and dense.

simultaneous data delivery and distance sensing (*i.e.,* ranging) is the fundamental basis.

This paper proposes to use ultra-wideband (UWB) radio technology to implement both networking and ranging for swarms. Because UWB is so time sensitive that it provides highly accurate transmission timestamp and reception timestamp for each data message exchanged between any pair of devices, by which the real-time distance between each pair can be constantly calculated and updated during data communications. Here, a timestamp for an event, either transmission or reception, is the time moment when the event is taking place.

This paper therefore focuses on designing a UWB swarm ranging protocol that achieves wireless data communication and swarm ranging simultaneously, where *swarm ranging* is a real-time mechanism that allows a device/robot to compute the distances to all its peer neighbors at the same time. Note that the ranged distances are fundamental for positioning and localization. Existing UWB-based distance ranging approaches do not consider dynamic and dense networks, which makes them different from ours.

We now summarize the major challenges as follows.

Large number. A swarm may contain a large number of devices and robots that must access a shared common wireless channel. Because of the broadcast nature of the wireless channel, inappropriate protocol may cause frequent transmission conflicts and thus prolonging a successful ranging operation and reducing its frequency. Also because of the large number of devices, the protocol must minimize the number of necessary messages for the ranging job and minimize the computation time for low-end devices and robots. Therefore, how to design a *simple yet efficient* UWB swarm ranging protocol is a challenging task.

High mobility. Very often, highly frequent ranging operations are desired for achieving high mobility. However, sometimes the devices and robots may also need to stay motionless as applications required. In this case, the ranging frequency should be reduced to save wireless channel resources. Besides, high mobility may cause the wireless communication unstable, such that the ranging messages get lost. Therefore, how to design an *adaptive and robust* UWB swarm ranging protocol to deal with high mobility problem is another challenging task.

Short distance. The distance between two devices/robots in a swarm may be short, so a device or robot may have many nearby neighbors to ranging with. However, the ranging message has a size limitation and cannot carry ranging information (timestamps) for all neighbors to compute distance at once. Besides, a good ranging protocol should be supportive and compatible with the existing networking and localization protocols. Therefore, how to design a *scalable and compatible* UWB swarm ranging protocol is also a challenge we need to deal with.

In summary, the objective of this paper is to design a UWB swarm ranging protocol for dynamic and dense networks that efficiently computes for each device or robot the distances towards all its neighbors. The contributions of this paper are summarized below.

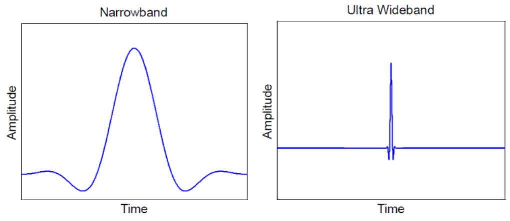
1) Our designed protocol seems to be a pioneering work that uses UWB technology for dynamic and dense networks to simultaneously achieve wireless ranging and data communication which has broad application prospects that can be used in various wireless networks, including robotic networks, body area networks, and so on.
2) Our protocol is efficient and easy to implement. It is very efficient because our protocol design takes advantage of the broadcast nature of wireless communication. It is easy to implement on all types of devices/robots, including the lower-end ones, because it is very simple that needs only one message type.
3) Our ranging protocol handles dynamic network smoothly, because it can adaptively adjust the ranging frequency in real-time according to the changing distance between the two devices in each pair. This protocol can also handle the message loss.
4) Our ranging protocol handles dense network smoothly. A bus boarding based scheme is designed to handle the case when a device/robot has too many neighbors that the size-limited ranging message cannot carry ranging information (timestamps) for all neighbors to compute distance at once.
5) We have implemented this protocol on Crazyflies, STM32 microcontroller powered micro drones, with only 192KB memory and onboard UWB wireless transceiver chips DW1000. Extensive real-world experiments are conducted to evaluate the proposed protocol on various performance aspects, with a total of 9 Crazyflie drones in a compact area. Our implementation is open-sourced at https://github.com/SEU-NetSI/crazyflie-firmware.

The rest of the paper is organized as follows. Section II introduces the basic knowledge on UWB and ranging protocol. Section III presents the detailed design of the swarm ranging protocol. Section IV gives the implementation details. Experiment results are discussed in Section V. Related work is summarized in Section VI. Section VII concludes this paper, discusses possible enhancements and practical implementation works in the future. Our preliminary work was reported in reference [8].

II. Preliminary

A. Ultra-Wideband (UWB) Radio Technology

UWB is a wireless transmission technology for propagating data at high bit rates over a wide frequency spectrum. One characteristic of UWB is that bits are transmitted using thin pulses - on the order of 1 ns in width. The differences between



(a) Narrowband (b) Wideband

Fig. 2. Compared with common narrowband radio technology, UWB propagates data using thin pulses, which are short and sharp, enough to provide accurate timestamps.

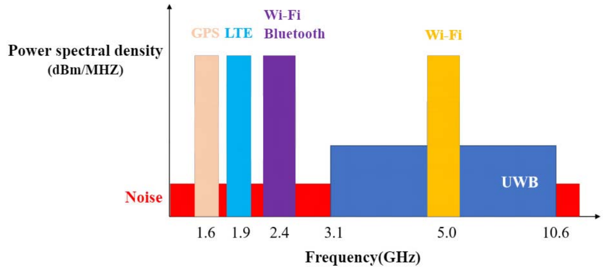


Fig. 3. The wide bandwidth and low spectral density enable UWB equipped devices to transmit data with low energy consumption.

common narrowband radio technology (such as WiFi, LTE) and UWB are shown in Fig. 2. Given the size and shape of the pulse, it is possible to measure the time of flight with great accuracy [9].

A typical UWB radio transceiver chip spans radio frequencies band from 3.5 GHz to 6.5 GHz and has a ranging accuracy of less than 10 cm [10]. As shown in Fig. 3, the wide bandwidth and low spectral density of UWB enable UWB devices to transmit data with low energy consumption. UWB also performs well in complex environments for its strong anti-multipath and penetration ability.

Utilizing the outstanding characteristics of UWB in positioning and data transmission, telecommunication companies are gradually laying out the UWB market. Apple has brought its self-developed UWB chips, namely U1, to consumer-grade products, such as the newest iPhone and Apple Watch. Companies such as Samsung and Xiaomi have also proposed their UWB solutions. Both Samsung and Apple have recently released UWB portable tracking devices in their product ecosystems, namely SmartTag Plus [11] and AirTag [12]. Because of its superior safety and high ranging accuracy, UWB radio technology is now gradually being used in more and more applications, such as car keys, motion capture, IoT device positioning and message interaction. The UWB precision ranging specification is recently defined in the latest IEEE Standard 802.15.4z-2020 [13]. It is foreseeable that UWB technology will be used in more and more scenarios in the future.

B. Double-Sided Two-Way Ranging (DS-TWR) Protocol

There is a brand new standardized ranging protocol defined in IEEE Standard 802.15.4z-2020 [13], namely double-sided two-way ranging (DS-TWR) protocol which we briefly introduce here.

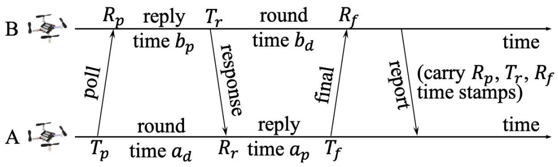


Fig. 4. The double-sided two-way ranging (DS-TWR) protocol.

As shown in Fig. 4, there are four types of messages, *i.e.,* *poll*, *response*, *final*, and *report* message, exchanged in order between the two sides, A and B, for each ranging operation. The transmission timestamp and reception timestamp for the first 3 messages are denoted as T_p, R_p, T_r, R_r, T_f, and R_f, respectively. At each side, the time duration from a reception time to the next transmission time is called a *reply* time, and the time duration from a transmission time to the next reception time is called a *round* time. Let us denote these durations for the two sides in Fig. 4 as follows.

$$a_d = R_r - T_p, b_p = T_r - R_p, b_d = R_f - T_r, a_p = T_f - R_r. \tag{1}$$

Let t_p be the time of flight (ToF), that is the radio signal propagation time from one side to the other side. Then, the two round time durations can be written as $a_d = 2t_p + b_p, b_d = 2t_p + a_p$. So, ToF can be calculated [10], [13] as

$$t_p = \frac{a_d b_d - a_p b_p}{a_d + b_d + a_p + b_p}. \tag{2}$$

By involving 4 parameters, Eq. (2) would have better error control than other equations such as $t_p = (a_d - b_p)/2$. Because each side has a clock offset caused by its crystal offsets, the actually calculated value for ToF can be expressed as follows, where e_A and e_B are clock errors of sides A and B, respectively.

$$\hat{t}_p = \frac{a_d(1+e_A) \times b_d(1+e_B) - a_p(1+e_A) \times b_p(1+e_B)}{a_d(1+e_A) + b_d(1+e_B) + a_p(1+e_A) + b_p(1+e_B)}.$$

Its deviation from the real true value is of Eq. (2) is

$$\hat{t}_p - t_p = \frac{e_A + e_B + 2e_A e_B}{2 + e_A + e_B} t_p. \tag{3}$$

It can be seen that if e_A and e_B are small, the deviation is small.

It is worth noting that in most operating systems of devices and robots, a transmission timestamp is available only after the message has been sent. As a result, an additional *report* message is needed to carry three related timestamps from one side to the other for ToF calculation.

C. DS-TWR Based on Token Ring

A simple extension of the DS-TWR protocol has been proposed [14], [15] to deal with many-to-many ranging operations, that use the token ring technique to control the ranging process. The basic idea is that all neighboring devices form

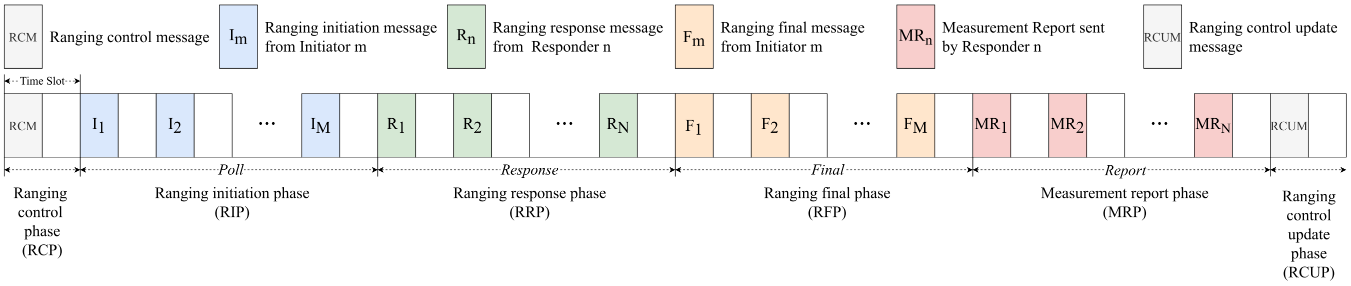

RCM Ranging control message
I_m Ranging initiation message from Initiator m
R_n Ranging response message from Responder n
F_m Ranging final message from Initiator m
MR_n Measurement Report sent by Responder n
RCUM Ranging control update message
Time Slot
RCM
I_1
I_2
...
I_M
R_1
R_2
...
R_N
F_1
F_2
...
F_M
MR_1
MR_2
...
MR_N
RCUM
Poll
Response
Final
Report
Ranging control phase (RCP)
Ranging initiation phase (RIP)
Ranging response phase (RRP)
Ranging final phase (RFP)
Measurement report phase (MRP)
Ranging control update phase (RCUP)

Fig. 5. The newly amended standard extends the DS-TWR protocol to many-to-many scenario with six ranging phases in a ranging round.

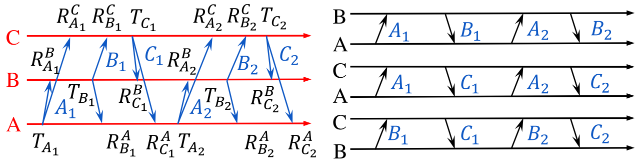
a ring and there is only one token. So, at any time there is only one device holding the token. It is the token holder, who initiates a DS-TWR protocol for each neighbor in turn, one by one. Once the token-holder has completed ranging for all neighbors, the token is passed to the next device in the ring.

Although this extension is simple to understand, it has some disadvantages. (1) Whenever two sides are exchanging messages, these messages can be heard by a number of neighbors because of the broadcast nature of the wireless communication. However, they are ignored, which makes the protocol inefficient. (2) All neighbors must know each other in advance in order to form a token ring. This imposes an additional neighbor discovery and expiration mechanism that makes the protocol poorly scalable. (3) Since the wireless communication is unreliable, messages may get lost. When the token gets lost, the entire ranging process stops. So the token must be monitored and recovered from loss, which would complicate its implementation.

D. Extension of DS-TWR in IEEE Standard 802.15.4z-2020

The newly amended IEEE standard 802.15.4z-2020 also includes many-to-many ranging protocol based on the DS-TWR protocol. In this protocol, the basic time unit is called ranging round, and the swarm ranging distances are updated once in every ranging round. More specifically, each ranging round consists of 6 ranging phases, including four phases dedicated to transmitting *poll*, *response*, *final*, and *report* messages. The time in each phase is further slotted and there are enough slots in each phase for each device to collect sufficient timestamps from all its neighbors to calculate ToF according to Eq. (2). More details are given in Fig. 5, from which we can conclude the new protocol is an improved DS-TWR because these four types of message play similar roles as in the DS-TWR protocol, but each needs more time slots because the initiating device may have many neighbors to ranging with.

Although this many-to-many ranging protocol from the new IEEE standard is a good extension to the DS-TWR, it lacks support for dynamic and dense networks. First, the media access method is based on pre-allocating time slots, which is unfriendly for dynamically join and leave neighbors. Because the time slot allocation is updated once a ranging round, a new neighbor will have to wait for the next ranging round to join the ranging operations. Second, a ranging round may take a very long time to complete for dense networks, because a device or robot may have many neighbors to ranging with, and each of them must be allocated multiple time slots. Therefore, the denser a network is, the longer a ranging round will take.


$R^C_{A_1}$ $R^C_{B_1}$ T_{C_1} $R^C_{A_2}$ $R^C_{B_2}$ T_{C_2}
C
$R^B_{A_1}$ B_1 C_1 $R^B_{A_2}$ B_2 C_2
B
T_{B_1} $R^B_{C_1}$ T_{B_2} $R^B_{C_2}$
A_1 A_2
A
T_{A_1} $R^A_{B_1}$ $R^A_{C_1}$ T_{A_2} $R^A_{B_2}$ $R^A_{C_2}$
B A_1 B_1 A_2 B_2 A
C A_1 C_1 A_2 C_2 A
C B_1 C_1 B_2 C_2 B

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

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

Fig. 6. An illustration of the new protocol for the three-side example.

The above brief survey and analysis for the current state of the art of UWB ranging protocol inspired us to design a new swarm ranging protocol that would improve on the currently known protocols.

E. The Basic Idea of a New Swarm Ranging Protocol

Recall that the DS-TWR protocol has four types of messages exchanged between two sides in order. After one side initiates a *poll* message, the two sides take turns to reply *response*, *final*, and *report* message. For the new swarm ranging protocol, we use only a single type of message, named the *ranging message*. We let each side periodically transmits a *ranging message* which is not responding to any particular incoming message. The new protocol will still use six timestamps to compute the ToF according to Eq. (1) and (2). In the following, we use a three-side toy example to explain how to accomplish ranging operations with only one type of messages. This example helps to understand the intrinsic logic behind the new protocol.

Fig. 6 shows the sequence of ranging messages transmitted by each side in the toy example, during the execution of the new protocol.

As illustrated by Fig. 6, three sides A, B, and C take turns to transmit six message, namely A_1, B_1, C_1, A_2, B_2, and C_2. Each message can be received by the other two sides because of the broadcast nature of wireless communication. Then every message generates three timestamps, *i.e.,* one transmission timestamp and two reception timestamps, as illustrated in Fig. 6(a). We can see that each pair has two rounds of message exchanges as shown in Fig. 6(b). Therefore, there are sufficient timestamps to calculate the ToF for each pair.

With each side transmitting only two messages, the distances of all three pairs can be calculated, which is much more efficient than the token ring based extension and the many-to-many DS-TWR extension in IEEE Standard 802.15.4z-2020. This observation inspires us to design a novel swarm ranging protocol. But, we have to deal with the challenges of *large number*, *high mobility* and *short distance* for the dynamic and dense network of devices and robots. In summary, the following questions must be answered.

Q1 (Simple yet efficient) How should we design the format of *ranging message* so that sufficient timestamps can be available to calculate the ToF?

Q2 (Simple yet efficient) How does the enlarged transmission period affect accuracy? We need to answer this question because in DS-TWR, a message is immediately replied, but in the new protocol, messages are sent periodically.

Q3 (Adaptive) How does high mobility affect the ranging accuracy? If the transmission frequency of *ranging message* stays low, but the devices or robots become moving faster, we need to know how the ranging accuracy will be affected?

Q4 (Adaptive) How often should the *ranging message* be broadcasted? When two devices or robots are far apart or moving slow, the ranging frequency can be low, but when they are close or moving fast, then it must be high to avoid collision. But what is the criterion?

Q5 (Robust) What should we do if a message gets lost or the ranging frequency mismatch each other that causes the unbalanced message exchange for two sides?

Q6 (Scalable) How should we handle dense swarms where a device has many neighbors? We need to answer this question because in this case a *ranging message* has no space to carry all the timestamps.

Q7 (Compatible) Can the new protocol support and be compatible with higher level protocols, such as optimized link state routing protocol and network localization algorithms?

III. A New Swarm Ranging Protocol

A. The Design of Ranging Message and the Main Framework

For notational consistency and clarity, let $A_k, k = 1, 2, \ldots$, denote the k^{th} message transmitted by device A, let T_{A_k} denote the transmission time of A_k, and $R^Y_{A_k}$ denote the time message A_k is received by Y, we sometimes use the simple notation R_{A_k} if no ambiguity arises.

Recall the three-side example in Fig. 6, let us focus on *ranging message* C_2. Upon receiving C_2, A and B should be able to calculate the ToF to C, respectively, because each of them has collected six (different) timestamps needed for this ToF calculation according to Eq. (1) and (2). More specifically, device A has collected T_{A_1}, $R^C_{A_1}$, T_{C_1}, $R^A_{C_1}$, T_{A_2}, $R^C_{A_2}$. Similarly, device B has collected T_{B_1}, $R^C_{B_1}$, T_{C_1}, $R^B_{C_1}$, T_{B_2}, $R^C_{B_2}$. Therefore C_2 must include the transmission timestamp of C_1 (available only after C_1 completed transmission), and reception timestamps of A_2 and B_2.

Definition 1 (Ranging Message): The ranging message *X_i is the i-th message broadcasted by device or robot X. It is defined to be*

$$\textit{Message } X_i = (X_i, T_{X_{i-1}}, M^x_i, v),$$

where X_i is the message identification, e.g., *sender X and sequence number i; $T_{X_{i-1}}$ is the transmission timestamp of previous message X_{i-1}; M^x_i is the set of reception timestamps of those messages sent by neighbors,* e.g., *$M^x_i = \{(A_2, R_{A_2}), (B_2, R_{B_2}), \ldots\}$ which should be sufficient for each neighbor to calculate the current ToF; v is the current velocity of X.*

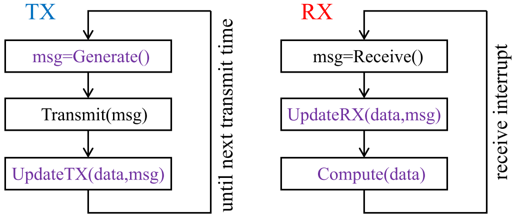


Fig. 7. The main framework of the swarm ranging protocol: the transmission (TX) part and the reception (RX) part.

The main framework of the swarm ranging protocol has two parts, *i.e.,* the transmission (TX) part and the reception (RX) part. Both parts are executed by each device X, as illustrated in Fig. 7. The TX part is executed once in a while. It consists of 3 steps, that is generate the message, broadcast the message, and update the ranging data. The RX part is executed once a message is received. It also consists of 3 steps, that is get the message, update the ranging data and compute the distance.

Procedures `Generate`, `UpdateTX`, `UpdateRX`, `Compute` and ranging data are keys to design the swarm ranging protocol.

B. Message Generating and Data Updating in a Simple Scenario

Let us start our protocol design with a simple scenario where there are a number of devices or robots, A, B, C, *etc*, in a short distance. Each one of them periodically transmits the *ranging messages* that can be heard by all others, and they broadcast the messages at the same pace. As a result, between any two consecutive message transmissions, a device or robot can hear messages from all others. The *ranging message* is designed to include all these reception timestamps along with their message identification. The pseudo code is presented in procedure `GenerateS`.

Procedure 1 `GenerateS`$(A_i, T_{A_{i-1}}, v)$

```
$M^x_i \leftarrow \emptyset$;
for each received message $Y_j$ since last transmission do
    $M^x_i \leftarrow M^x_i \cup (Y_j, R_{Y_j})$
end
return Message$(A_i, T_{A_{i-1}}, M^x_i, v)$
```

Let us focus on the message exchanges process between one of the pairs, A and Y, where Y can be B, C, *etc*. The first few message exchanges are shown in Fig. 8, where

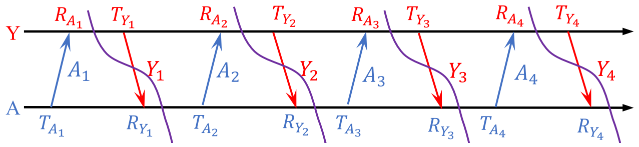

Y
A
R_{A_1} T_{Y_1} R_{A_2} T_{Y_2} R_{A_3} T_{Y_3} R_{A_4} T_{Y_4}
A_1 Y_1 A_2 Y_2 A_3 Y_3 A_4 Y_4
T_{A_1} R_{Y_1} T_{A_2} R_{Y_2} T_{A_3} R_{Y_3} T_{A_4} R_{Y_4}

Fig. 8. The first few message interaction between A and Y (can be B, C, *etc*).

TABLE I
THE RANGING TABLE DATA STRUCTURE FOR A SIMPLE CASE

Y side	$R_p = R_{A_{i-1}}$	$T_r = T_{Y_{j-1}}$	$R_f = R_{A_i}$	
A side	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_{j-1}}$	$T_f = T_{A_i}$	$R_e = R_{Y_j}$

each delivered message has a transmission timestamp and a reception timestamp. From the point of view of A, upon receiving message Y_1, timestamps R_{A_1}, R_{Y_1} become known according to procedure `GenerateS` of Y. Since message Y_2 carries its last transmission time T_{Y_1}, its latest reception time R_{A_2} and other reception timestamps (not for A). Upon receiving Y_2, 7 timestamps are available (on the left of the 2nd purple line), while the first 6 timestamps can be used to calculate the ToF by Eq. (1) and (2). Similarly, new ToF can be calculated upon receiving Y_3 and Y_4.

In general, a new ToF can be calculated upon any receipt of *ranging messages*, and only a small portion of timestamps are needed for the computation. We therefore propose a ranging table data structure as Table I that stores the latest 7 timestamps. Each time a *ranging message* is received, the ToF is calculated and the ranging table is updated.

Assume A_{i-1} is the last transmitted message and Y_{j-1} is last received message from Y, then in the ranging table data structure (Table I), T_p, R_p and R_r are initially known for the new round. After delivering message A_i, T_f is updated to T_{A_i}. The formal steps of updating are in procedure `UpdateTX-S`. Upon receiving Y_j, its reception timestamp R_{Y_j} can be updated as R_e. Meanwhile, Y_j brings two Y side timestamps, $T_{Y_{j-1}}$ and R_{A_i}, thus T_r and R_f can be updated. The formal steps of updating are in procedure `UpdateRX-S`.

Procedure 2 `UpdateTX-S`(tables, msg)

```
for each existing table in tables do
    Update its $T_f$.
end
```

Procedure 3 `UpdateRX-S`(tables, msg)

```
if $msg$ is from neighbor $Y$ then
    if table(AY) is not existing then
        Initialize table(AY);
    end
    Update $T_r, R_f$ and $R_e$ of table(AY);
end
```

Note that A may have many neighbors, it thus has many ranging tables to maintain, one for each neighbor. Every time a device or robot receives a ranging message from a new neighbor, it will initialize a new ranging table for this neighbor. For a transmitted message, the transmission timestamp should be updated into every ranging table, while the reception timestamp for a message from an existing neighbor, should be updated into the corresponding ranging table for the neighbor.

After the ranging table is updated, the ToF can be calculated as in procedure `ComputeS`. Note that table updating in Line 3 and 4 is to prepare for the next round of ToF calculation.

Procedure 4 `ComputeS`(Table(AY))

```
1 if Table(AY) is incomplete then return $\varnothing$;
2 Compute ToF by Eq. (1) and (2) using data in table(AY);
3 $R_p \leftarrow R_f, T_p \leftarrow T_f, R_r \leftarrow R_e$;
4 $T_r \leftarrow \varnothing, R_f \leftarrow \varnothing, T_f \leftarrow \varnothing, R_e \leftarrow \varnothing$;
5 return $ToF$
```

Hereby, Q1 is answered, *i.e.,* timestamps to be included in the next ranging message are in procedure `GenerateS`, and the steps to update the ranging table are in procedure `UpdateTX-S` and `UpdateRX-S`.

We now turn to Q2, *i.e.,* does the enlarged transmission period affect accuracy? In our swarm ranging protocol, a message is transmitted periodically, therefore the reply delay (duration between the reception and transmission timestamp) is obviously larger than that of DS-TWR, where a message is replied immediately. However, according to Eq. (3), the accuracy of the computed ToF is only related to the offset error of the two devices' crystal, not the reply delay. So, Q2 is answered.

C. Design of Adaptive Ranging Protocol

Some important questions still remain open: how to set an appropriate *ranging message* transmission period for a fast moving device or robot (Q3 and Q4)? We define the *ranging period* P to be the time period between two consecutive ranging message transmission. It is clear that a short ranging period results in high frequency ranging,[1] which is helpful for fast moving devices and robots. However, the short ranging period may cause the wireless channel to be occupied by too many ranging messages, which leads to message confliction and reduce the data communication throughput on the same channel. When two devices or robots are far apart or they move quite slow, the ranging frequency should be set low.

The *ranging period* P should be adapted to the velocity and distance. We investigate such adaption in a high speed but large ranging period scenario. As in Fig. 9, assume A is moving towards Y with relative velocity v. Let P be the ranging period for both A and Y, during which, A moves vP distance. Assume A transmit A_{i-1}, receive Y_{j-1}, transmit A_i and receive Y_j sequentially at d_1, d_2, d_3 and d_4 distances towards Y. So t_{p_1}, t_{p_2}, t_{p_3} and t_{p_4} are durations for the wireless signal to travel these distance, respectively. Define $t_\triangle$ be the time for the wireless signal to travel vP distance, hence $t_\triangle c = vP$, where c is the light speed. Within a transmission

[1] We use *ranging* and *ToF calculation* interchangeably in this paper.

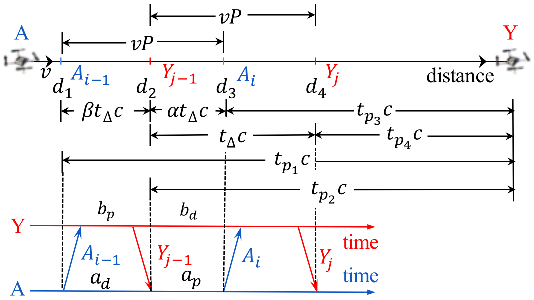


Fig. 9. A high velocity but large ranging period scenario.

period, there must be a ranging message received from the other side. Let the reception time divides a transmission period into a $\beta : \alpha$ ratio, where $\alpha + \beta = 1$. Notations and their relationships are detailed in Fig. 9.

Hence, we have $t_{p_1} = t_{p_2} + \beta t_\Delta$, $t_{p_3} = t_{p_2} - \alpha t_\Delta$, and $t_{p_4} = t_{p_2} - t_\Delta$. Therefore, $a_d = b_p + 2t_{p_2} + \beta t_\Delta$ and $b_d = a_p + 2t_{p_2} - \alpha t_\Delta$. Because the processing time is far larger than the signal propagation time, so $a_d \approx b_p$ and hence $b_p : a_p = \beta : \alpha$ As a result, we get the computed ToF as shown in Eq. (4), as shown at the bottom of the next page.

Note that the last approximation is because $a_p, b_p \gg t_{p_2}, t_\Delta$. The actual ToF towards Y at location d_4 should be

$$t_p^{actual} = t_{p_4} = t_{p_2} - t_\Delta = t_p^{computed} - \frac{vP}{c}. \tag{5}$$

If we want to bound the error by a constant e_0,

$$\frac{|t_p^{actual} - t_p^{computed}|}{t_p^{actual}} \leq e_0,$$

then, we get $\frac{t_\Delta}{t_{p_2}} \leq \frac{e_0}{1-e_0}$. Since $vP = t_\Delta c$, we have

$$P \leq \frac{e_0}{1 - e_0}\frac{d_2}{v}. \tag{6}$$

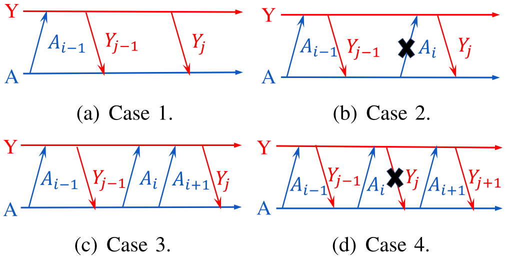
Eq. (6) serves as a guideline to determine the *ranging period* P with given velocity v and computed distance d_2.

However, in practice, both sides have clock offsets caused by crystal offsets. As mentioned earlier, let the crystal error be e_A and e_Y respectively. Hence, the actual ranging period is $P_A = P(1 + e_A)$ for A and $P_Y = P(1 + e_Y)$ for Y. During which period, A moves vP_A distance and vP_Y distance respectively. Hence $t_\Delta^A c = vP_A$. As previously assumed, the reception time divides P_A into a $\beta : \alpha$ ratio, where $\alpha + \beta = 1$. Using the same derivation steps, we shall have $t_{p_1} = t_{p_2} + \beta t_\Delta^A$ and $t_{p_3} = t_{p_2} - \alpha t_\Delta^A$. Therefore, $a_d^A = b_p^Y + 2t_{p_2} + \beta t_\Delta^A$ and $b_d^Y = a_p^A + 2t_{p_2} - \alpha t_\Delta^A$. It is worth noting that both sides have their own clock offsets, so $a_d^A = a_d(1 + e_A)$, which rule is also used for a_p^A, b_d^Y and b_p^Y.

Hence, when considering the effect of crystal error, the actual computed ToF is Eq. (7), as shown at the bottom of the next page.

The last approximation in Eq. (7) is correct, because $a_p, b_p \gg t_{p_2}, t_\Delta$ and e_A, e_Y is relatively very small.

In summary, even if the crystal error is introduced, Eq. (6) still serves as a guideline to determine the ranging period P.



(a) Case 1.

(b) Case 2.

(c) Case 3.

(d) Case 4.

Fig. 10. The ranging period mismatch and message loss cases.

Finally, we can see from Eq. (6) that the smaller distance, the shorter ranging period; meanwhile, the faster velocity, the shorter ranging period, which answers Q3 and Q4.

D. Handle Ranging Period Mismatch and Message Loss

According to the previous analysis, Eq. (6), for any pair of device or robot, there is a ranging period, which may be different from other pair, because the distance d_2 and velocity v may be different. In other words, one device or robot may have different ranging period for each of its neighbors. For example, 'far and slow' neighbor wants large ranging period and thus low ranging frequency, while 'close and fast' neighbor needs small ranging period and thus high ranging frequency. To handle these differences, a simple way is to choose the tightest ranging frequency requirement (smallest ranging period) and set it for all neighbors that will satisfy all requirements.

$$P_A = \min_{Y \in \text{neighbors}} \{\frac{e_0}{1 - e_0}\frac{d_{AY}}{v_{AY}}\}, \tag{8}$$

where d_{AY} and v_{AY} are the distance and speed.

As a result, each device or robot may have its own ranging period (frequency requirement) by Eq. (6). Therefore, a direct question follows: what if the ranging counterparts have a different ranging period or what if messages are lost in transmission (Q5)? Both questions concern the unbalanced message exchange, which we summarize as four cases in Fig. 10. In Case 1, A receives more than it transmits. In Case 2, one message from A is lost. In Case 3, A transmits more than it receives. In Case 4, one message from Y is lost. Our swarm ranging protocol must be able to handle all cases.

In both Case 1 and Case 2, message Y_j will not carry any reception timestamp for A, because no message received since Y_{j-1} transmitted. Therefore, in the ranging table, R_f will be absent after the receive update, as the gray cell in Table II. For such two cases, our solution is to discard the timestamps of Y_{j-1} and A_i (if it exists), as shown in the lower part of Table II, so that the next round of ranging can be continued.

Case 3 is easy to handle. We always update/overwrite the transmission timestamp T_f of the ranging table whenever a message is transmitted.

In Case 4, since message Y_j is lost, after A_{i+1} is transmitted, the transmission timestamp T_f is overwritten by $T_{A_{i+1}}$. At the time when A receives message Y_{j+1}, there exists a mismatch. On the Y side, Y_j is the latest transmitted message, so timestamp T_{Y_j} is carried in message Y_{j+1}. This timestamp

TABLE II

HANDLE THE RANGING PERIOD MISMATCH AND MESSAGE LOSS CASE 1 AND 2

Y side	$R_p = R_{A_{i-1}}$	$T_r = T_{Y_{j-1}}$	$R_f =$	
A side	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_{j-1}}$	$T_f = T_{A_i}$ or $T_f =$	$R_e = R_{Y_j}$

$\Downarrow$

Y side	$R_p = R_{A_{i-1}}$	$T_r =$	$R_f =$	
A side	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_j}$	$T_f =$	$R_e =$

TABLE III

HANDLE THE MISMATCH AND LOSS CASE 4

Y	$R_p = R_{A_{i-1}}$	$T_r = T_{Y_j}$	$R_f = R_{A_{i+1}}$	
A	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_{j-1}}$	$T_f = T_{A_{i+1}}$	$R_e = R_{Y_{j+1}}$

$\Downarrow$

Y	$R_p = R_{A_{i+1}}$	$T_r =$	$R_f =$	
A	$T_p = T_{A_{i+1}}$	$R_r = R_{Y_{j+1}}$	$T_f =$	$R_e =$

is updated into the ranging table such that $T_r = T_{Y_j}$. However, on the A side, because Y_{j-1} is the last received message, so the ranging table keeps $R_r = R_{Y_{j-1}}$. Then, there is a mismatch on message index, as shown in gray cells of Table III. In this case, our solution is to discard the timestamps of A_{i-1}, Y_{j-1} and Y_j, as shown in the lower part of Table III, so that the next round of ranging can be continued.

We summarize these steps by procedure UpdateRX-M.

Please note that, if the table AY is incomplete, procedure ComputeS will not compute the ToF according to the **if** statement in Line 1. Hereby, Q5 is answered.

Procedure 5 UpdateRX-M(tables, msg)

```
UpdateRX-S(tables, $msg$);
if $msg$ is from neighbor $Y$ then
    if $R_f == \varnothing$ in table(AY) then
        $R_r \leftarrow R_e, T_r \leftarrow \varnothing, T_f \leftarrow \varnothing, R_e \leftarrow \varnothing$ for table(AY);
    else if index($T_r$) $\neq$ index($R_r$) in table(AY) then
        $R_p \leftarrow R_f, T_p \leftarrow T_f, R_r \leftarrow R_e$ for table(AY);
        $T_r \leftarrow \varnothing, R_f \leftarrow \varnothing, T_f \leftarrow \varnothing, R_e \leftarrow \varnothing$ for table(AY);
    end
end
```

E. Handle Dense Neighbors

By procedure GenerateS, all reception timestamps since last transmission are included in the ranging message. The ranging period that controls how often the ranging message is transmitted, is set to the smallest ranging period of all neighbors according to Eq. (8). As a result, devices and robots exchange ranging messages at a high frequency, and the number of reception timestamps included in every ranging message is large, especially when there are dense neighbors.

However, the message size is limited by the hardware or standards. Once there are too many neighbors, some timestamps must be discarded ultimately. It is critical to answer Q6: how to select timestamps to fit the message capacity so as to be scalable to dense neighbors?

In order to make a good decision on such selection for procedure GenerateS, one nature idea is to set priorities, *i.e.,* the lower ranging frequency required, the lower priority. In other words, 'far and slow' neighbors' timestamps are the first to drop. Yet, our practice has shown, this simple priority system causes the starvation problem, that is the high ranging frequency neighbors always get chances to deliver their timestamps, while the low frequency one get little chances because of their low priorities.

We want to improve the priority system design. We introduce our idea by a comprehensive example of the bus boarding problem. A group of passengers need to take the same bus at a bus station, so they share the bus seats. Every passenger takes the bus once a while, repeating every repeat-riding time. Note that, the repeat-riding time may be different from passenger to passenger. We want to arrange the limited bus seats for the

$$
\begin{aligned}
t_p^{computed} &= \frac{a_d \times b_d - a_p \times b_p}{a_d + b_d + a_p + b_p} \\
&= \frac{t_{p_2}(2a_p + 2b_p + 4t_{p_2} - \alpha t_\Delta + \beta t_\Delta) + t_{p_2} t_\Delta(\beta - \alpha) + (\beta a_p - \alpha b_p)t_\Delta - \alpha\beta t_\Delta^2)}{4t_{p_2} + 2a_p + 2b_p + t_\Delta(\beta - \alpha)} \\
&= t_{p_2} + \frac{t_{p_2} t_\Delta(\beta - \alpha) + (\beta a_p - \alpha b_p)t_\Delta - \alpha\beta t_\Delta^2}{4t_{p_2} + 2a_p + 2b_p + t_\Delta(\beta - \alpha)} \approx t_{p_2},
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
\hat{t}_p^{computed} &= \frac{a_d^A \times b_d^Y - a_p^A \times b_p^Y}{a_d^A + b_d^Y + a_p^A + b_p^Y} \\
&= \frac{t_{p_2}(2a_p^A + 2b_p^Y + 4t_{p_2} - \alpha t_\Delta^A + \beta t_\Delta^A) + t_{p_2} t_\Delta^A(\beta - \alpha) + (\beta a_p^A - \alpha b_p^Y)t_\Delta^A - \alpha\beta(t_\Delta^A)^2)}{4t_{p_2} + 2a_p^A + 2b_p^Y + t_\Delta^A(\beta - \alpha)} \\
&= t_{p_2} + \frac{t_{p_2} t_\Delta^A(\beta - \alpha) + (\beta a_p^A - \alpha b_p^Y)t_\Delta^A - \alpha\beta(t_\Delta^A)^2}{4t_{p_2} + 2a_p^A + 2b_p^Y + t_\Delta^A(\beta - \alpha)} \\
&= t_{p_2} + \frac{t_{p_2} t_\Delta(\beta - \alpha)(1 + e_A) + (\beta a_p(1 + e_A) - \alpha b_p(1 + e_Y))t_\Delta(1 + e_A) - \alpha\beta t_\Delta^2(1 + e_A)^2}{4t_{p_2} + 2a_p(1 + e_A) + 2b_p(1 + e_Y) + t_\Delta(\beta - \alpha)(1 + e_A)} \approx t_{p_2},
\end{aligned}
\tag{7}
$$

TABLE IV

THE IMPROVED RANGING TABLE DATA STRUCTURE

Y side	R_p	T_r	R_f	P	t_n
A side	T_p	R_r	T_f	R_e	t_s

passengers such that everyone takes turns to ride the bus and satisfy everyone's repeat-riding time as much as possible.

Our basic idea is simple. We maintain a next want-ride time for each passenger in the group. Once the bus arrived at the station, whoever waits at the station gets on the bus according to their next want-ride time. The passenger with the most urgent next want-ride time is the next to board the bus and take a seat, and so on. Once all seats are taken, the bus leaves, and all the leftover passengers wait for the next bus. Every boarded passenger updates his next want-ride time to one repeat-riding time later from the current time. All the leftover passengers have their next want-ride time unchanged. In this way, when the next bus arrives, the leftover passengers from the last bus and the small repeat-riding time passengers have priorities for boarding. Moreover, we stop maintaining the next want-ride time for a passenger after he does not show up at the station for a very long time. We call this scheme the bus boarding scheme.

The bus boarding scheme is designed to improve procedure `GenerateS`. The readers may have already guessed that, the bus represents the ranging message, the passengers represent the reception timestamps from neighbors. As a result, a passenger boarding the bus represents the reception timestamp for a neighbor is selected to be included in the ranging message to be broadcasted next.

We now improve the ranging table data structure as shown in Table. IV. There are three new notations compared to Table I, *i.e.,* P, t_n, t_s, where P denotes the newest ranging period for neighbor Y, *i.e.,* the repeat-riding time for passenger Y; t_n denotes the *next (expected) delivery time* for neighbor Y, *i.e.,* the next want-ride time for passenger Y; and t_s denotes the expiration time for neighbor Y, *i.e.,* the stop maintaining time for passenger Y. Note that for each neighbor Y, A maintains a ranging table(AY).

The ranging period is updated right after the ToF is calculated, as in procedure `Compute`.

Procedure 6 `Compute`(Table(AY))

```
$ToF$ =ComputeS(Table(AY));
Update $P$ by Eq. (6) for table(AY);
return $ToF$
```

The *next (expected) delivery time* t_n is updated once a ranging message containing the reveive timestamp from neighbor Y is delivered, *i.e.,* t_n should be one period P later than the current time. While the expiration time t_s is updated once a message from neighbor Y is received, *i.e.,* t_s should be the expiration time later than the current time. The detailed pseudo code can be found in procedure `UpdateTX` and `UpdateRX`.

Procedure 7 `UpdateTX`(tables, msg)

```
UpdateTX-S(tables, $msg$);
for each $Y$ whose reception timestamp is carried in $msg$ do
    $t_n \leftarrow t_{current} + P$ for table(AY);
end
```

Procedure 8 `UpdateRX`(tables, msg)

```
UpdateRX-M(tables, $msg$);
if $msg$ is from neighbor $Y$ then
    $t_s \leftarrow t_{current} + T_{expiration}$ for table(AY);
end
```

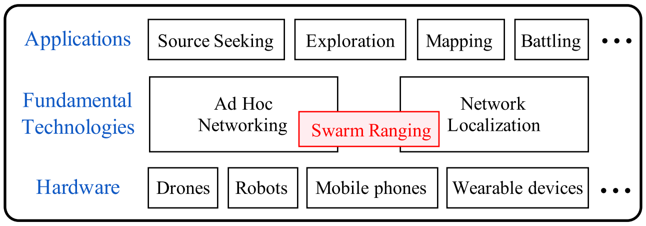


Fig. 11. Swarm ranging supports ad hoc networking and network localization that are fundamental for any swarm application.

With the ranging table improved and appropriately maintained, we now focus on how to generate a ranging message according to the *next (expected) delivery time* t_n and the expiration time t_s from all ranging tables. Assume that the *ranging message* has a capacity to carry at most m reception timestamps. The selection principle is simple: for the neighbors that are not yet expired, we sort them by the *next (expected) delivery time* so that they follow the chronological order, and choose m most urgent neighbors. A *ranging message* carrying reception timestamps for these m neighbors is generated and transmitted. Then, the m next delivery times are updated by procedure `UpdateTX`. The procedure repeats and the next *ranging message* is generated at the nearest *next (expected) delivery time*. Procedure `Generate` presents the details.

Procedure 9 `Generate()`

```
for each existing table(AY) do
    if $t_{current} > t_s$ then delete table(AY);
end
Sort all tables by $t_n$ in ascending order;
$M_i^x \leftarrow \emptyset$;
for each neighbor $Y$ from top $m$ tables do
    $M_i^x \leftarrow M_i^x \cup (Y, R_Y)$
end
return Message$(X_i, T_{X_{i-1}}, M_i^x, v)$
```

With the help of the above design, our protocol now handles the dense neighbor, which answers Q6.

F. Support for Higher Level Protocols and Algorithms

For any successful swarm application, for example, a team of indoor drones searching for given targets, a swarm of small robots exploring and mapping an unknown indoor

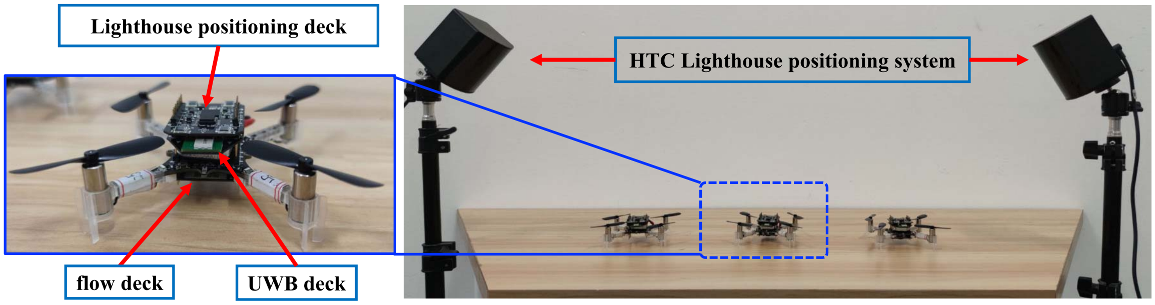
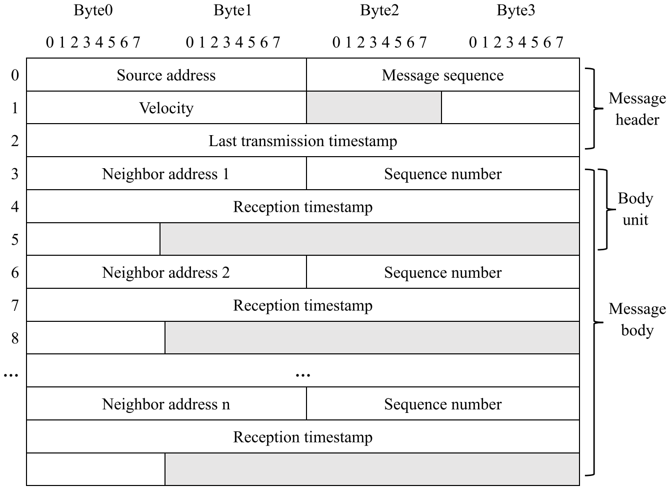

Lighthouse positioning deck
HTC Lighthouse positioning system
flow deck
UWB deck

Fig. 12. The hardware used in our experiments, including Crazyflie drones and HTC Lighthouse positioning system.

environment, an army of legged robot dogs battling in a deep forest, two fundamental technologies are 1) low latency ad hoc networking and 2) real time network localization. An architecture is given in Fig. 11.

Swarm ranging supports ad hoc networking and network localization simultaneously. (1) In ad hoc networks, some high level routing protocols require to maintain a list of active neighbors. Our swarm ranging protocol already maintains such a list (by the expiration time t_s), which can be used directly. Moreover, some routing protocols broadcast probe messages periodically, *e.g.,* the `HELLO` message in OLSR [16], which can be combined with our *ranging message*. (2) In network localization, a network of nodes is used to aid in localizing its members, especially with distance information between each other [15], [17]. Our swarm ranging protocol provides the fundamental distance information to support higher level localization algorithms. Hereby, Q7 is answered.



	Byte0	Byte1	Byte2	Byte3	
	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	
0	Source address		Message sequence		Message header
1	Velocity				
2	Last transmission timestamp				
3	Neighbor address 1		Sequence number		Body unit / Message body
4	Reception timestamp				
5					
6	Neighbor address 2		Sequence number		
7	Reception timestamp				
8					
...	...				
	Neighbor address n		Sequence number		
	Reception timestamp				

Fig. 13. Our implemented ranging message format.

IV. Implementation

We implement our proposed swarm ranging protocol on Crazyflies, STM32 microcontroller powered micro drones, with only 192KB memory and onboard UWB wireless transceiver chips DW1000. In our experiment, data from drones is collected by a laptop through Crazyradio PA, a communication device working at 2.4GHz band. A flow deck is equipped for automatic flight control and height measurement.

In some experiments that involves distance ground truth, an HTC Vive Lighthouse positioning system is used, which is an optical indoor position system providing millimeter-level position data treated as the ground truth. An additional lighthouse deck needs to be installed onto the drone to get the positioning data. Fig. 12 shows the hardware for our experiment. There are lighthouse base stations on the left and right sides. On the desk are three Crazyflie drones, each equipped with a lighthouse deck, a UWB deck on the top and a flow deck on the bottom.

All drones share the same UWB channel for the purpose of broadcasting. We set the data rate at 6.8 Mbps and use 128bit of preamble code. Fig. 13 shows our implemented version of the ranging message. In our implementation, a ranging message consists of a message header and a message body. In the message header, the source address field is used to identify the sender of the ranging message. The message sequence field stores the sequence number of this message. The velocity field and the last transmission time field are the corresponding data involving in our proposed protocol. The message body consists of a set of body units, one for each neighbor. One body unit carries the identification and timestamp data of the ranging message from the neighbor declared in the neighbor address field. Recall the ranging message formal definition, in Definition 1, the source address field and the message sequence field match X_i. The last transmission time field matches $T_{X_{i-1}}$. M_i^x is implemented as the message body and the velocity field corresponds to v.

V. Experiment

This section evaluates the swarm ranging protocol in various performance terms through experiments.

To reduce the ranging message collision probability, we randomize the period P by dividing it into two parts, *i.e.,* $p + w$. The base p is a given fixed real number $p < P$. The random w is a random number following a uniform distribution $w \in U(0, W)$. We name W as the random window size. So we have $\bar{P} = E(p + w) = p + W/2$.

A. Ranging Period and Accuracy

To evaluate the impact of ranging period on ranging accuracy, we put two drones stopped on two separate chairs with

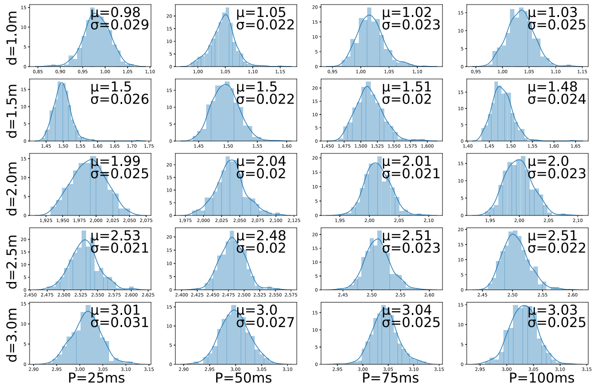


Fig. 14. The impact of ranging period on ranging accuracy.

TABLE V

MESSAGE RECEPTION RATIO AND RANGING RATIO

Ranging Pairs	Reception Count	Reception Ratio	Ranging Count	Ranging Ratio
AB	5591	93.18%	4473	74.55%
AC	5578	92.97%	4441	74.02%
AD	5568	92.80%	4430	73.83%

a clear line of sight, and more than 1 meter away from any wall. We disable the ranging period adaption and set $W = 0$. We study the ranging accuracy under various ranging period P from 25ms to 100ms with step 25ms, and various distance d from 1.0m to 3.0m with step 0.5m. At each setting, the ranging results follow some random distribution. Its probability density function is plotted in Fig. 14. It is clear to conclude that the ranging period P do not affect ranging accuracy at various distance.

B. Message Reception Ratio and Ranging Ratio

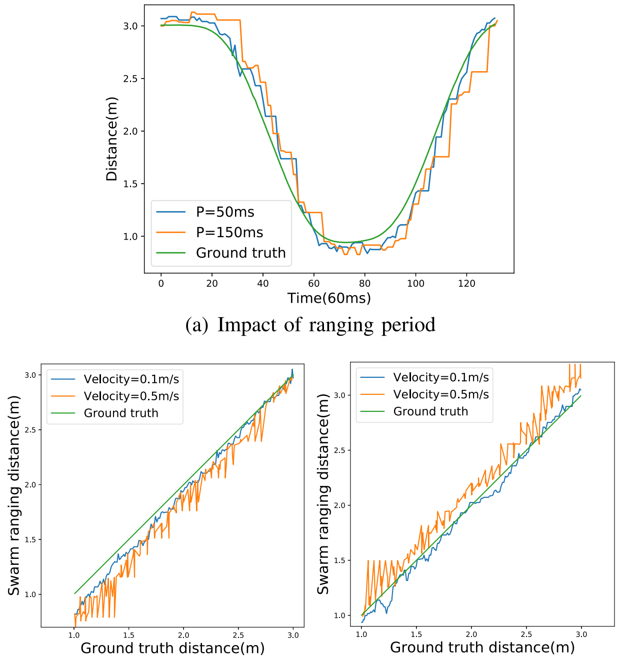
This subsection utilizes 4 drones to form a small swarm to evaluate the *message reception ratio* and *ranging ratio*. The 4 drones, namely A, B, C, and D, are placed at a close distance statically to guarantee a good wireless channel. We set for all drones the average ranging period $\bar{P} = 50$ms ($p = 30$ms and $W = 40$ms). Each drone transmits a total of 6000 ranging messages.

We define the *message reception ratio* as the ratio of the received message count to the transmitted message count, and the *ranging ratio* as the ratio of successful ranging calculation counts to the message transmitted. Table V shows the counts and ratios recorded by drone A. It is clear that the average *message reception ratio* is over 90% and the average *ranging ratio* is over 70%. The reason for the low *ranging ratio* is because the ranging period is randomized, which causes period mismatches randomly, *i.e.,* Case 1 and 3 from Fig. 10, and thus reduces the *ranging ratio*.

C. Velocity and Ranging Accuracy

This subsection evaluates the impact of velocity v on ranging accuracy. The HTC Lighthouse indoor positioning system is used to provide distance ground truth.

According to Eq. (5), both ranging period P and drone speed v affect the ranging accuracy. Therefore, we study them separately.



(a) Impact of ranging period

(b) Impact of velocity and direction (moving away).

(c) Impact of velocity and direction (moving towards).

Fig. 15. Ranging accuracy with different velocity and ranging period.

First, the impact of the ranging period is evaluated using two drones A and B. Drone A hovers at a fixed location. Drone B first flies towards it, starting from a location 3 meters away, both at the same height, 100 cm. B then stops at 1 meter distance from A, and later flies back to its starting location. The flight speed for B is set to $v = 0.5$m/s. We repeat twice with the ranging period $P = 50$ms and 150ms respectively. The ranging results are shown in Fig. 15(a). It is clear that when P becomes larger, the accuracy drops. We also observe that the ranging results are delayed compared to the ground truth, and the larger P, the bigger delay, which is consistent with our theoretical analysis in Eq. (5).

Second, the impact of flight speed is investigated by the same settings. The differences are we now fix the ranging period to $\bar{P} = 50$ms ($p = 45$ms and $W = 10$ms), but set two flight speeds for drone B, 0.1m/s and 0.5m/s. We separate two flight directs, moving towards and moving away, as in Fig. 15(b)(c). The faster flight speed, the fewer sampling ranging data collected, because fewer time is spent on fly. To get enough sample points, drone B is set to fly 5 times at a speed of 0.5m/s. As can be seen in Fig. 15(b)(c), the ranging results are more stable for $v = 0.1$m/s than 0.5m/s. Moreover, (b)(c) also reveal the same conclusion from (a), the ranging has a delay.

D. Performance for Mismatched Ranging Period

In this experiment, there are 4 drones, namely A, B, C, and D. We set the average ranging period $\bar{P} = 30$ms, 40ms, 50ms, and 60ms for A, B, C, and D, respectively, where the random window size is set $W = 40$ms for all drones. Fig. 16 shows the ranging counts recorded by drone A in a duration of 200s. We can conclude that our swarm ranging protocol handles the ranging period mismatching smoothly, that is, the shorter

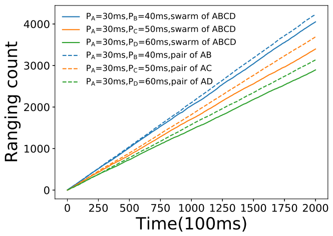


Fig. 16. Performance for mismatched ranging period.

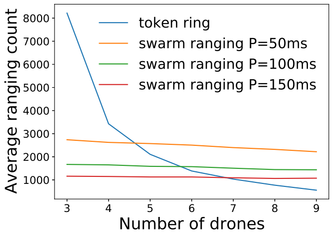


Fig. 17. Comparison with token ring based ranging.

period its neighbor has, the more ranging occurred. Moreover, the ranging count increases when the pair is taken out of the swarm and range without interference.

E. Comparison With Token Ring Based Ranging

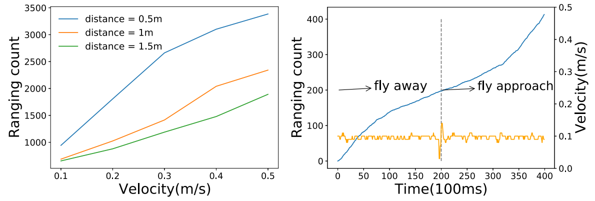
We compare our swarm ranging protocol to the token ring based ranging algorithm. We vary the average ranging period $\bar{P}$, from 50ms to 150ms, while keeps the random window size $W = 80$ms. We vary the numbers of drones, from 3 to 9. Fig. 17 shows the ranging counts recorded by drone A for 200s.

Any point in Fig. 17 is the average successful ranging count by drone A with its neighbors. We can see that the performance of token ring algorithm decreases dramatically with the growth of participants, because it executes sequentially. While the performance of swarm ranging decreases slightly because the probability of message collision increases as the number increases. When there are more than 5 drones, our swarm ranging protocol outperforms the token ring algorithm. When 9 drones participate, the average number for a drone to successfully range with another drone by our protocol is about 5 times higher than that by the token ring algorithm.

F. Adaptability of the Ranging Period

In this subsection, we evaluate the adaptability referred to Eq. (6) that both velocity v and distance d can affect the period P. Instead of setting fixed p and W, we allow the ranging period P be adjusted according to Eq. (6). The HTC lighthouse system is used for positioning and distance measuring.

We conduct two experiments. In the first experiment, drone A hovers at a fixed location while drone B circles around drone A at a fixed velocity v. Fig. 18(a) shows the ranging count recorded by drone B in a duration of 100s. We vary the distance d from 0.5m to 1.5m with step 0.5m and drone B's velocity v from 0.1m/s to 0.5m/s with step 0.1m/s. As shown in Fig. 18(a), the ranging count increases as velocity increases. Furthermore, the ranging count drops when the distance increases. In the second experiment, drone A and B hover at a fixed location, close to each other. Then drone B files away from drone A at a speed of 0.1m/s and returns after flying 20 seconds. Fig. 18(b) shows the ranging count recorded by drone A and the speed of drone B. We can see that the growth of the ranging count slows down as drone B flies away and quickens as drone B approaches. All results from the two experiments validate the protocol's adaptability of the ranging period.



(a) Adaptive to velocity. (b) Adaptive to distance.

Fig. 18. The ranging period is adaptive to velocity and distance.

G. Performance of the Bus Boarding Scheme for Dense Swarms

The IEEE Standard 802.15.4z-2020 limits the packet size to 127 bytes [13]. As the packet payload, the ranging message has a size limitation of 100 bytes, because the packet header has to be subtracted. As a result, the number of the body unit (one for each neighbor) is limited to 7, as shown in Fig. 13.

In this experiment, we evaluate the performance of the bus boarding scheme proposed for the swarm ranging protocol when neighbors are dense in a swarm. We put 11 drones close to each other and limit each ranging message to carry at most 7 body units, so it is not always possible to include all received timestamps for all neighbors in the ranging message. To evaluate the performance of the bus boarding scheme in selecting neighbors for the next ranging message, we compare it with two simple baseline methods. The first baseline method does not consider priority, *i.e.,* whenever a selection must be made, the neighbors are chosen purely based on their identification (ID) number. The second baseline method proposes a simple priority system, *i.e.,* the higher required ranging frequency, the higher priority, and neighbors are selected by this order. We conduct two experiments, comparing with these two baselines respectively.

In the first experiment, we set the average ranging period $\bar{P} = 50$ms (W = 0ms) for each drone and record the average ranging count in a duration of 200s. The results are shown in Fig. 19. We can see that, by the first baseline method, drones with ID greater than 8 have much less ranging count. This is because, these drones get very fewer chances to perform ranging since the ranging message can only carry ranging information (body unit) for 7 neighbors. On the contrary, our bus boarding scheme guarantees all neighbors have fair chances to ranging. And we can see from the figure that there is no large differences in ranging counts for all drones.

In the second experiment, we set different ranging period for different neighbors. We set $\bar{P} = 50$ms for drone 1, 65ms for drone 2, 80ms for drone 3, and so on. By the simple priority system used in the second baseline, the drones with larger ranging period (smaller required ranging frequency)

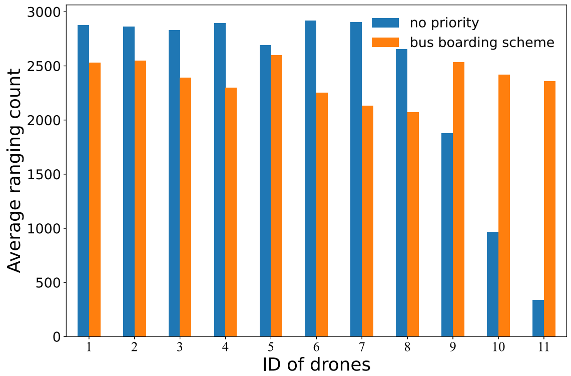


Fig. 19. The bus boarding scheme balances the allocation of ranging resources.

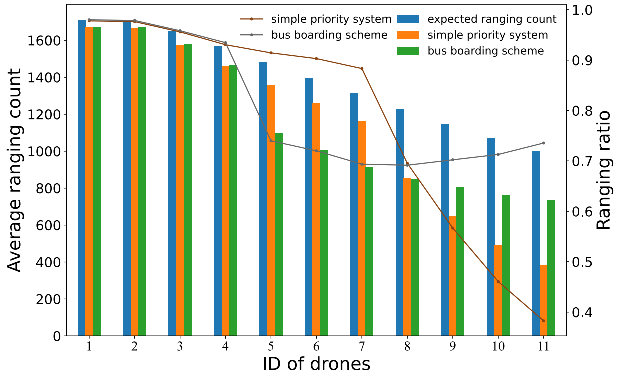


Fig. 20. The bus boarding scheme guarantees fair chances for each drone to ranging.

have lower priorities to compete for ranging messages. The results are shown in Fig. 20, in which the expected ranging count refers to the theoretical upper bound of the ranging count, *i.e.,* no message loss and a very large packet size that allows all neighbors to range.

The ranging ratio (right y-axis) is the ratio between the measured value and the expected value. We can see from the figure that, in the simple priority system, the ranging ratio of low ranging frequency drone drops sharply as the ID number grows larger than 7. As a comparison, the ranging ratios are over 69% for all drones in our solution, which indicates our bus boarding scheme ensures fair chances for all neighbors to compete ranging resources with each other.

H. Demonstration Experiment

We conduct a collision avoidance experiment to test the real time ranging accuracy. A demonstration video can be found at https://www.youtube.com/watch?v=jT3mPQvHadM.

In this experiment, 8 Crazyflie drones hover at a height of 70cm in a compact area less than 3m by 3m. While the ninth Crazyflie drone is manually controlled to fly into this area. We set the average ranging period $\bar{P} = 100$ms ($p = 60$ms and $W = 80$ms) and the maximum ranging message body unit (one for each neighbor) to be 8. As shown in the demo video and Fig. 21, thanks to the swarm ranging protocol, a drone detects the coming drone by ranging distance and lowers its height to avoid collisions once the distance is small than threshold 30cm.

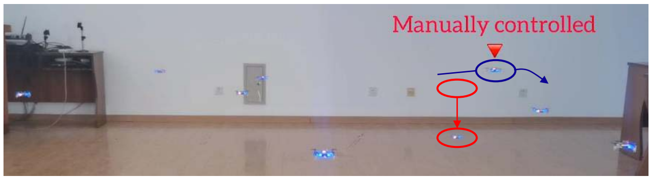


Fig. 21. Crazyflie drones avoiding collision when flying compactly.

VI. Related Work

UWB is being used widely for indoor localizations [18]–[24]. Tiemann *et al.* [18] and Corbalan *et al.* [19] design TDOA based algorithms for ranging which require time synchronization. While Xu *et al.* [20] fuse visual, inertial, and UWB information for aerial swarm localization. At the same time, Cao *et al.* [21] locate a robot with only a single UWB anchor (base station). Moreover, Poulose *et al.* [22] use a long short-term memory (LSTM) network to predict user position in an indoor scene by UWB. Yu *et al.* [23] propose a less environment-dependent and a prior knowledge-independent NLOS identification and mitigation method for ranging and also develop a rule to select appropriate NLOS ranges for location estimation. Macoir *et al.* [24] design a UWB based localization system that uses infrastructure anchor nodes. Guizar *et al.* [25] investigate the impact of MAC scheduling strategies on ranging accuracy of UWB in motion capture systems. Courtay *et al.* [26] propose a platform for indoor positioning and motion tracking in wireless body area networks. In addition to UWB, there are other techniques that have been used for localization in swarm networks. For example, Rivard *et al.* [27] develop a relative positioning system based on time-of-flight estimation of ultrasonic pulses. Also, De Silva *et al.* [28] propose an ultrasonic-based 3D sensor node for multirobot localization. While, Wu *et al.* [29] propose a method to precisely locate devices in robotic networks with ZigBee anchor nodes. At the same time, McGuire *et al.* [30] apply bluetooth for relative localization and collision avoidance in micro air vehicle swarms; Khanh *et al.* [31] propose a Wi-Fi based indoor positioning and navigation method using a cloudlet-based cloud computing system.

Recently, concurrent ranging is becoming an emerging trend in ultra-wideband research community [32]–[35]. Corbalan *et al.* [32], [33] propose to use the channel impulse response to discriminate the individual times of arrival of the overlapping of replies for the same request, which is called the concurrent ranging. Heydariaan *et al.* [35] later extends the idea to reflection resilient. While Stocker *et al.* [34] extends it to support ranging for unlimited number of tags.

There are also a few works focus on UWB swarm ranging for large numbers [19], [34], [36] or for high mobility [37], [38]. Corbalan *et al.* [19] and Stocker *et al.* [34] focus to locate countless tag by TDOA or by concurrent ranging. Their work is dedicated to the anchor-tag model, not applicable for swarm scenarios. Cao *et al.* [36] propose a novel Time Division Multiple Access (TDMA) algorithm that can quickly schedule the use of the UWB medium among a large network consisted of UWB-tagged nodes. Risset *et al.* [37] investigate the UWB swarm ranging problem for rapid movements, with only two UWB tags. Liu *et al.* [38] proposes an effective system framework of INS/UWB integrated positioning for autonomous indoor mobile robots.

UWB is also been used for data communication [39], [40]. Mohammadmoradi *et al.* [39] propose to simultaneous ranging and communication in UWB networks. Vecchia *et al.* [40] investigate the concurrent transmission problem for UWB.

The newly amended IEEE Standard 802.15.4z-2020 [13] propose solutions for multi-node ranging based on SS-TWR or DS-TWR. These solutions always involve multiple ranging phases and roles (*i.e.,* controller, controlee, initiator, responder) to define a ranging procedure. A ranging process is composed of multiple ranging rounds, where a ranging round is composed of multiple phases. Each phase is a period of sufficient duration for a certain role to achieve a certain purpose. For example, the ranging control phase is usually the first phase in a ranging round so that the controller can send a control message that decides ranging parameters. Though broadcast is applied, scheduled phases require a centralized ranging process. These solutions are not only complex but also difficult to configure.

As a conclusion, none of the related work focus on design a ranging protocol specially for dynamic and dense networks.

VII. Conclusion and Future Works

This paper proposes a UWB swarm ranging protocol, specially designed for dynamic and dense robotic or device networks. The basic idea is to design the *ranging message* which is broadcasted periodically. Timestamps are carried by this message so that the distance can be calculated. Our swarm ranging protocol is *simple yet efficient* because there is only one single type of message; it is *adaptive and robust* because the ranging period adapts to the ranging pair's speed and distance, and message loss is handled appropriately; it is *scalable and compatible* because a bus boarding scheme is designed to handle dense neighbors and higher level networking and localization protocols and algorithms are supported. Finally, this protocol is implemented on Crazyflies, STM32 microcontroller powered micro drones, with only 192KB memory and onboard UWB wireless transceiver chips DW1000. Thanks to the swarm ranging protocol, a total of 9 Crazyflie drones can automatically avoid collision when flying in a compact space.

In the future, more efforts can be made to improve the current design and implementation of our swarm ranging protocol. First, data mining techniques can be applied to utilize the large amount of ranging data generated from the network to achieve higher ranging accuracy. Second, implement this protocol in a more generalized way to interact with smartphones, wearable devices, and more to form larger networks and design data fusion algorithm to combine multiple types of sensor data to reduce ranging errors. Third, implement this protocol on advanced hardware with faster CPU and larger memory to support larger networks with hundreds of devices or robots. Fourth, the proposed ranging resource allocation algorithm can be adjusted or improved depending on the requirement of the application tasks. Lastly, some possible enhancement efforts can be put on the trade-off between packet loss rate and ranging frequency in the dynamic and dense networks.

Acknowledgment

The authors would like to thank Prof. Xiaojun Shen from the University of Missouri-Kansas City (UMKC), the anonymous referees, and the editors for their constructive comments and valuable suggestions, which have helped improve the quality and presentation of the paper.

References

[1] *Crazyflie 2.1—Bitcraze Store*. Accessed: Jul. 22, 2021. [Online]. Available: https://www.bitcraze.io/2019/02/the-crazyflie-2-1-is-here/

[2] *RoboMaster EP—DJI*. Accessed: Jul. 22, 2021. [Online]. Available: https://www.dji.com/robomaster-ep

[3] *Boston Dynamics Launches Commercial Sales of SPOT Robot*. Accessed: Jul. 22, 2021. [Online]. Available: https://www.bostondynamics.com/press-release-spot-commercial-launch

[4] Z. Zhakypov, K. Mori, K. Hosoda, and J. Paik, "Designing minimal and scalable insect-inspired multi-locomotion millirobots," *Nature*, vol. 571, no. 7765, pp. 381–386, Jul. 2019.

[5] J. Yu, D. Jin, K.-F. Chan, Q. Wang, K. Yuan, and L. Zhang, "Active generation and magnetic actuation of microrobotic swarms in bio-fluids," *Nature Commun.*, vol. 10, no. 1, pp. 1–12, Dec. 2019.

[6] Y. Yang and M. A. Bevan, "Cargo capture and transport by colloidal swarms," *Sci. Adv.*, vol. 6, no. 4, Jan. 2020, Art. no. eaay7679.

[7] K. N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Sci. Robot.*, vol. 4, no. 35, Oct. 2019, Art. no. eaaw9710.

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

[9] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*. London, U.K.: Pearson, 2012.

[10] *DW1000 User Manual*, Decawave Ltd, Dublin, Ireland, 2017.

[11] *SmartTag Plus—Samsung*. Accessed: Jun. 25, 2021. [Online]. Available: https://www.samsung.com/us/mobile/mobileaccessories/phones/galaxy-a-smart-tag-1-pack-black-ei-t7300bbegus/

[12] *AirTag—Apple*. Accessed: Jun. 25, 2021. [Online]. Available: https://www.apple.com/airtag/

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

[14] B. Broecker, K. Tuyls, and J. Butterworth, "Distance-based multi-robot coordination on pocket drones," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6389–6394.

[15] S. Li, M. Coppola, C. De Wagter, and G. C. H. E. de Croon, "An autonomous swarm of micro flying robots with range-based relative localization," 2020, *arXiv:2003.05853*.

[16] (2003). *Optimized Link State Routing Protocol (OLSR)*. Internet Requests for Comments, RFC Editor, RFC 3626. [Online]. Available: https://tools.ietf.org/html/rfc3626

[17] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. V. Poor, "Network operation strategies for efficient localization and navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, Jul. 2018.

[18] J. Tiemann, Y. Elmasry, L. Koring, and C. Wietfeld, "ATLAS FaST: Fast and simple scheduled TDOA for reliable ultra-wideband localization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 2554–2560.

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

[20] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, "Decentralized visual-inertial-UWB fusion for relative state estimation of aerial swarm," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8776–8782.

[21] Y. Cao, C. Yang, R. Li, A. Knoll, and G. Beltrame, "Accurate position tracking with a single UWB anchor," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2344–2350.

[22] A. Poulose and D. S. Han, "UWB indoor localization using deep learning LSTM networks," *Appl. Sci.*, vol. 10, no. 18, p. 6290, Sep. 2020.

[23] K. Yu, K. Wen, Y. Li, S. Zhang, and K. Zhang, "A novel NLOS mitigation algorithm for UWB localization in harsh indoor environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 686–699, Jan. 2019.

[24] N. Macoir *et al.*, "UWB localization with battery-powered wireless backbone for drone-based inventory management," *Sensors*, vol. 19, no. 3, p. 467, Jan. 2019.

[25] A. Guizar, A. Ouni, C. Goursaud, N. Amiot, and J.-M. Gorce, "Impact of MAC scheduling on positioning accuracy for motion capture with ultra wideband body area networks," in *Proc. 9th Int. Conf. Body Area Netw.*, 2014, pp. 1–8.

[26] A. Courtay, M. Le Gentil, O. Berder, A. Carer, P. Scalart, and O. Sentieys, "*Zyggie*: A wireless body area network platform for indoor positioning and motion tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[27] F. Rivard, J. Bisson, F. Michaud, and D. Letourneau, "Ultrasonic relative positioning for multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 323–328.

[28] O. D. Silva, G. K. I. Mann, and R. G. Gosine, "An ultrasonic and vision-based relative positioning sensor for multirobot localization," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1716–1726, Mar. 2015.

[29] X. Wu, H. Wu, B. Xin, and C. Chen, "Precise coordinated localization for swarm robots via multidimensional scaling and wireless sensor networks," in *Proc. IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, vol. 1, Dec. 2021, pp. 1–6.

[30] K. McGuire, M. Coppola, C. de Wagter, and G. de Croon, "Towards autonomous navigation of multiple pocket-drones in real-world environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 244–249.

[31] T. T. Khanh, V. Nguyen, X.-Q. Pham, and E.-N. Huh, "Wi-Fi indoor positioning and navigation: A cloudlet-based cloud computing approach," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, pp. 1–26, Dec. 2020.

[32] P. Corbalán and G. P. Picco, "Concurrent ranging in ultra-wideband radios: Experimental evidence, challenges, and opportunities," in *Proc. EWSN*, 2018, pp. 55–66.

[33] P. Corbalán and G. P. Picco, "Ultra-wideband concurrent ranging," *ACM Trans. Sensor Netw.*, vol. 16, no. 4, pp. 1–41, Nov. 2020.

[34] M. Stocker, B. Großwindhager, C. A. Boano, and K. Römer, "SnapLoc: An ultra-fast UWB-based indoor localization system for an unlimited number of tags: Demo abstract," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2019, pp. 61–72.

[35] M. Heydariaan, H. Mohammadmoradi, and O. Gnawali, "R3: Reflection resilient concurrent ranging with ultra-wideband radios," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 1–8.

[36] Y. Cao, C. Chen, D. St-Onge, and G. Beltrame, "Distributed TDMA for mobile UWB network localization," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13449–13464, Sep. 2021.

[37] T. Risset, C. Goursaud, X. Brun, K. Marquet, and F. Meyer, "UWB ranging for rapid movements," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2018, pp. 1–8.

[38] J. F. Liu, J. X. Pu, L. F. Sun, and Z. S. He, "An approach to robust INS/UWB integrated positioning for autonomous indoor mobile robots," *Sensors*, vol. 19, no. 4, p. 950, Feb. 2019.

[39] H. Mohammadmoradi, M. Heydariaan, and O. Gnawali, "SRAC: Simultaneous ranging and communication in UWB networks," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 9–16.

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

Feng Shan (Member, IEEE) received the Ph.D. degree in computer science from Southeast University, Nanjing, China, in 2015. He was a visiting student with the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University. His research interests include the areas of Internet of Things, wireless networks, swarm intelligence, and algorithm design and analysis.

Haodong Huo (Graduate Student Member, IEEE) received the B.Sc. degree in computer science from Henan University, Kaifeng, China, in 2021. He is currently pursuing the master's degree in cyber science and security with Southeast University. His research interests include swarm intelligence, UAV networking, and algorithm design.

Jiaxin Zeng received the B.E. degree in computer science from the Hefei University of Technology, Hefei, China, in 2016, and the M.E. degree in computer technology from Southeast University, Nanjing, China, in 2021. He is currently a Software Engineer with China UnionPay Company Ltd.

Zengbao Li received the B.Sc. degree from the Anhui University of Technology, Anhui, China, in 2019, and the M.Sc. degree from Southeast University, Nanjing, China, in 2022. His research interests include wireless networking protocol and algorithm design.

Weiwei Wu (Member, IEEE) received the B.Sc. degree in computer science from the South China University of Technology (SCUT) in 2006 and the joint Ph.D. degree in computer science from the City University of Hong Kong (CityU) and the University of Science and Technology of China (USTC) in 2011. He went to the Mathematical Division, Nanyang Technological University (NTU), Singapore, for post-doctoral research in 2012. He is currently a Professor with the School of Computer Science and Engineering, Southeast University, China. His research interests include optimizations and algorithm analysis, crowdsourcing, reinforcement learning, game theory, wireless communications, and networks economics.

Junzhou Luo (Member, IEEE) received the B.Sc. degree in applied mathematics and the M.S. and Ph.D. degrees in computer networks from Southeast University, China, in 1982, 1992, and 2000, respectively. He is a Full Professor with the School of Computer Science and Engineering, Southeast University. His research interests are next generation networks architecture, networks security, cloud computing, and wireless LAN. He is a member of the IEEE Computer Society and the Co-Chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design. He is a member of the ACM and the Chair of ACM SIGCOMM, China.