Ultra-Wideband Swarm Ranging

Feng Shan*, Jiaxin Zeng†, Zengbao Li‡, Junzhou Luo*, Weiwei Wu*
* School of Computer Science and Engineering, Southeast University, Jiangsu, Nanjing 210096, China.
† SEU-Monash Joint Graduate School, Southeast University, Jiangsu, Suzhou 215123, China.
‡ School of Cyber Science and Engineering, Southeast University, Jiangsu, Nanjing 210096, China.
Emails: {shanfeng, jiaxinzeng, lizengbao, jluo, weiweiwu}@seu.edu.cn

***Abstract*—Nowadays, aerial and ground robots, wearable and portable devices are becoming smaller, lighter, cheaper, and thus popular. It is now possible to utilize tens and thousands of them to form a swarm to complete complicated cooperative tasks, such as searching, rescuing, mapping, and battling. A swarm usually contains a large number of robots or devices, which are in short distance to each other and may move dynamically. So this paper studies the dynamic and dense swarms. The ultra-wideband (UWB) technology is proposed to serve as the fundamental technique for both networking and localization, because UWB is so time sensitive that an accurate distance can be calculated using timestamps of the transmit and receive data packets. A UWB swarm ranging protocol is designed in this paper, with key features: simple yet efficient, adaptive and robust, scalable and supportive. This swarm ranging protocol is introduced part by part to uncover its support for each of these features. It is implemented on Crazyflie 2.1 drones, STM32 microcontrollers powered aerial robots, with onboard UWB wireless transceiver chips DW1000. Extensive real world experiments are conducted to verify the proposed protocol with a total of 9 Crazyflie drones in a compact area.**

***Index Terms*—UWB, Swarm, Ranging, Protocol Design, Networking, Two way ranging**

I. INTRODUCTION

With the rapid development of electronic manufacturing industry, more and more aerial and ground robots, wearable and portable devices are becoming commercially available. For example, 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], smart phones and smart watches from various vendors are being constantly released and upgraded.

Since these robots and devices are becoming smaller in size, lighter in weight, and cheaper in price, thus popular, it is now possible to utilize tens and thousands of them to form a robot and device swarm, and complete complicated tasks by cooperating with each other. For example, a team of indoor drones search for given targets, a swarm of small robots explore and map an unknown indoor environment, an army of legged robot dogs battle in a deep forest. Compared with a full functionality large single robot, a swarm of small robots and devices has the advantages of higher fault tolerant, more flexibility in deployment size and numbers, rapid deployment speed. As a result, the robot and device swarm is a current research trend, and publications are emerging in top journals such as *Nature* and *Science* [4], [5], [6], [7].

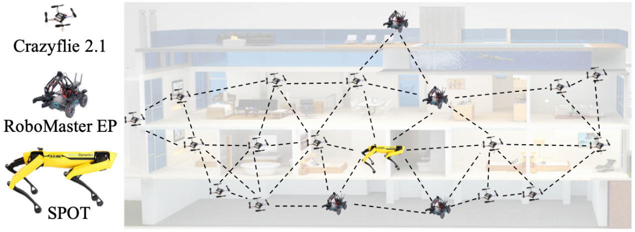


Fig. 1: Robot and device swarms are dynamic and dense.

We conclude three important features for robots and devices in such a swarm: *large number*, *high mobility* and *short distance*. First, according to the task complexity, tens and thousands of robots and devices may be deployed to collaborate. Second, micro drones, wheeled and legged robots, wearable and portable devices carried by human, are all capable of moving fast as required. Third, since these robots and devices are small in size, they can cooperate within short distances in order to complete complicated tasks. In a conclusion, robot and device swarms in the upcoming future applications are expected to be dynamic and dense.

A successful dynamic and dense swarm application requires low latency communication and real-time localization. When there is no outside supportive infrastructures, its is critical to preform ad hoc networking and in-swarm relative localization.

This paper proposes to use ultra-wideband (UWB) radio technology to implement both networking and localization for swarms. UWB is so time sensitive that an accurate distance can be calculated using timestamps of the transmit and receive data packets. In other words, wireless communicating and distance ranging can be achieved simultaneously, noting that the ranged distances are fundamental of localization. This paper focuses on designing a UWB ranging protocol for a dynamic and dense swarm.

We summarize the challenges as follows.

Large number. A swarm contains a large number of robots and devices, and because of the broadcast nature of the wireless ranging message, they access the same wireless channel. Inappropriate protocol may cause ranging messages conflicting and thus affect ranging frequency, while complicated protocol requires high computation which is precious for low end robots and devices. Therefore, how to design a *simple yet efficient* UWB ranging protocol is a challenge.

High mobility. High frequency ranging is desired for high mobility. But the robots and devices may also stay motionless as required. In case two robots or devices are far apart and barely move, the ranging frequency should be reduced to save precious wireless channel resources. Besides, high mobility may cause the wireless channel unstable, thus the ranging packets are subject to loss. How to design an *adaptive and robust* UWB ranging protocol is another challenge.

Short distance. The distance between two robots or devices in a swarm may be short, so a robot or device may have dense neighbors. However, the ranging message has a size limitation to carry ranging information (timestamps) for every neighbor. Besides, there already exist some networking and localization protocols. A good ranging protocol should be supportive and compatible with them. Therefore, how to design a *scalable and compatible* UWB ranging protocol is the third challenge.

Therefore, the objective of this paper is to design a UWB ranging protocol for a dynamic and dense swarm of robots and devices. Contributions of this paper are summarized below.

1) To the best of our knowledge, we are amount the first to apply UWB technology in dense and dynamic swarms, so as to simultaneously support wireless ranging and data communication.
2) We design a *simple yet efficient* UWB ranging protocol. There is only one single message type which is easy to implement, and the design takes advantage of the broadcast nature of wireless ranging message, thus is much more efficient within swarms.
3) We design an *adaptive and robust* UWB ranging protocol. The ranging frequency between the two ranging counterparts changes according to the relative velocity and distance. Packet loss and frequency mismatch is handled appropriately.
4) We design a *scalable and compatible* UWB ranging protocol. A rotation scheme is designed to handle the case of carrying too many neighbor information in a single ranging message. Our protocol is shown to be compatible with higher level networking protocols such as OLSR.
5) This protocol is implemented on Crazyflie 2.1 drones, STM32 microcontrollers powered aerial robots, with onboard UWB wireless transceiver chips DW1000. Extensive real world experiments are conducted to verify the proposed protocol with a total of 9 Crazyflies in a compact area.

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.

II. PRELIMINARY

A. Ultra-wideband (UWB) technology

UWB is a wireless communication technology for propagating data at high bit rates over a wide frequency spectrum. The bits are propagated using thin pulses - on the order of 1 ns in width.

A typical UWB radio transceiver chip spans radio frequency band from 3.5 GHz to 6.5GHz, and has a ranging accuracy of less than 10 cm [8]. The newest iPhone 11 (at the time of writing) has already been equipped with UWB chips [9].

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

There already exists a ranging protocol within IEEE Standard 802.15.4-2011 [10], which is later improved by industry giant Decawave [11], [8]. We refer to the improved ranging protocol as double-sided two-way ranging (DS-TWR) protocol, which we briefly introduce here. As shown in Fig. 2, there are four types of message, *i.e., poll*, *response*, *final*, and *report* message, exchanging between the two sides, A and B. We define their transmit and receive timestamps are T_p, R_p, T_r, R_r, T_f, and R_f, respectively. We define the reply and round time duration for the two sides as follows

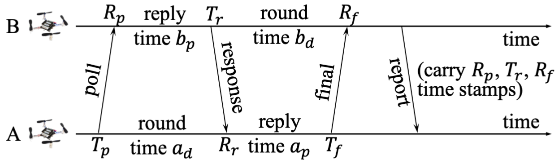


Fig. 2: The double-sided two-way ranging (DS-TWR) protocol.

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

which is shown in Fig. 2. Let t_p be the time of fly (ToF), namely radio signal propagation time. 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 [11], [8] as

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

However, both sides have clock offsets caused by crystal offsets. Let the crystal error be e_A and e_B respectively. Hence, our actual computed ToF is

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

Its deviation from the real true value is

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

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 robots and devices, a transmit 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. Ranging based on token ring

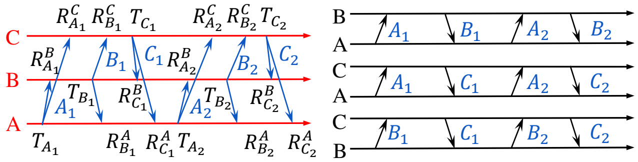
The DS-TWR protocol is designed for the one-to-one ranging. With the increasing needs for the ranging amongst multiple sides, a simple extension of the DS-TWR protocol has been proposed [12]. The token ring technique is adopted to control the ranging process. The basic idea is that all the sides form a ring and there is only one token, so at any time there is at most one side holding the token, who is allowed to initialize DS-TWR processes for all its neighbor in turn, one by one. Once the token-holding side has completed ranging, the token is passed to the next side.

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 is lack of efficiency. (2) All neighbors must be known before forming a token ring. This imposes an additional neighbor discovery and expiration mechanism must be designed, which is lack of scalability. (3) Since the wireless communication is unreliable, packets may be lost. When the token is lost, the ranging stops. So the token must be monitored and recovered from loss, which is complicated to implement.

This has inspired us to design a novel swarm ranging protocol from scratch.

D. The basic idea of swarm ranging

Recall that the DS-TWR protocol has four types of messages, and the two sides interact by replying a received message. For the new swarm ranging protocol, we want it as simple as possible, so we use a single type of message, named the *ranging* message. Each side periodically transmits a *ranging* message instead of reply to any received message. However, the ToF calculation needs six timestamps according to Eq. (1) and (2), how to implement ranging with only *ranging* messages? A three-sides toy example in Fig. 3 provides a good motivation and inspiration.



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

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

Fig. 3: A three-sides example inspires the design of swarm ranging protocol.

In Fig. 3, 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 transmit and two receive timestamps, as illustrated in Fig. 3(a). We can see that each pair has two rounds of message exchanges as shown in Fig. 3(b). Therefore, there are sufficient timestamps to calculate the ToF for each pair.

With each side transmitting only two messages, all three pairs can be ranged, which is much more efficient than the token ring based extension. 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 swarm of robots and devices. Hereby, the following questions must be answered.

Q1 (Simple yet efficient) How to design the *ranging* message so that sufficient timestamps are carried? Six timestamps are needed for the ranging calculation, three from each side. Which timestamps should be included in the next *ranging* message to be broadcasted?

Q2 (Simple yet efficient) Does the enlarged transmit period affect accuracy? In DS-TWR, a message is immediately replied, but in swarm ranging, messages are sent periodically. Does the enlarged reply time affect ranging accuracy?

Q3 (Adaptive) How does high mobility affect the ranging accuracy? If the *ranging* message is not broadcasted quite often but the robots or devices move quite fast, is the ranging accuracy affected?

Q4 (Adaptive) How often should the *ranging* message be broadcasted? When two robots or devices 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 to do if a message is lost or the ranging frequency mismatch each other so that the two sides broadcast unbalanced number of messages?

Q6 (Scalable) How to handle dense neighbors? In case there are too many neighbors that a ranging message can not carry all the timestamps, a good selection must be made.

Q7 (Compatible) Does the swarm ranging protocol supports or compatible with higher level protocols, such as OLSR routing protocol and network localization algorithms?

III. DESIGN OF SWARM RANGING PROTOCOL

A. The ranging *message and the main framework*

Recall the three-sides example in Fig. 3, 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. Both A and B needs six (different) timestamps for calculation according to Eq. (1) and (2). So C_2 must include the transmit timestamp of C_1 (available only after C_1 completed transmission), and some receive timestamps, *e.g.,* A_2 and B_2 receive time.

We now come to the formal definition.

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

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

where X_i is the message identification, e.g., *sender and sequence number; $T_{X_{i-1}}$ is the transmit timestamp of X_{i-1},* i.e., *the last sent message; RxM is the set of receive timestamps and their message identification,* e.g., $RxM = \{(A_2, R_{A_2}), (B_2, R_{B_2})\}$*; v is the current velocity of X.*

In the main framework of the swarm ranging protocol, there are two parts, *i.e.,* the transmit (TX) related part and

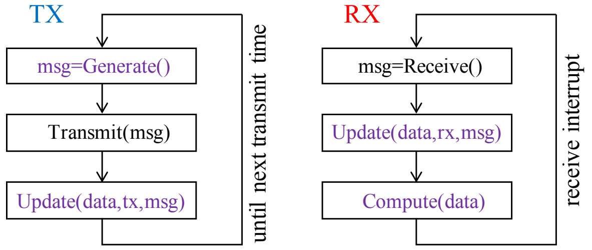


Fig. 4: The main framework of the swarm ranging protocol with two parts: the transmit (TX) part and the receive (RX) part.

the receive (RX) related part, as illustrated in Fig. 4. The TX part repeats once in a while: procedure `Generate()` produces the message msg, which is then broadcasted by procedure `Transmit()`, and then `Update()` is invoked to update the ranging data. The RX part executes every time a *ranging* message is received: procedure `Receive()` returns the message msg, which is used in `Update()` to update the ranging data, and the ToF is calculated by `Compute()` according to the updated ranging data.

The `Generate()`, `Update()`, `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 robots or devices, A, B, C, *etc*, in a short distance. Each one of them transmits a message can be heard by all others, and they broadcast *ranging* messages at the same pace. As a result, between any two consecutive message transmission, a robot or device can hear messages from all others. The *ranging* message is designed to include all these receive timestamps along with their message identification. The pseudo code is presented in procedure `GenerateS()`.

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

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

Let us focus on the message interaction between one of the pairs, A and Y, where Y can be B, C, *etc*, because they are equal. The first few message interaction is shown in Fig. 5, where each message has a transmit and a receive timestamp. From the point of view of A, upon receiving message Y_1, timestamps $T_{A_1}, R_{A_1}, R_{Y_1}$ are known according to `GenerateS`. Since message Y_2 carries its last transmit time T_{Y_1}, its latest receive time R_{A_2} and other receive timestamps not for A. Upon receiving Y_2, 7 timestamps is available (on 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 A_3 and A_4.

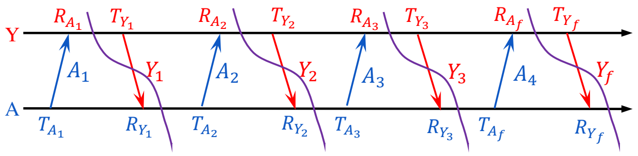


Fig. 5: The first few message interaction between A and Y (can be B, C, *etc*).

In general, a new ToF can be calculated upon any receipt of *ranging* messages, and only a small portion of timestamps are needed for computation. We therefore propose a ranging 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 data 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 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 be T_{A_i}. Upon receiving Y_j, its receive 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 `UpdateS`. Note that A may have many neighbors, it thus has many ranging tables to maintain, one for each neighbor. The transmit timestamp should be updated into every table, while the receive timestamp should be updated into the very table for the sender.

TABLE I. The ranging 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	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_{j-1}}$	$T_f = T_{A_i}$	$R_e = R_{Y_j}$

Procedure 2: `UpdateS`(tables, case, msg)

```
if case=='tx' then
    for each existing table do
        Update its $T_f$.
    end
else if case=='rx' then
    if $msg$ is from neighbor $Y$ then
        Update $T_r, R_f$ and $R_e$ of table(AY);
    end
end
```

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

Procedure 3: `ComputeS`(Table(AY))

```
1 if Table(AY) is incomplete then return ∅;
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. We now turn to Q2, *i.e.,* does the enlarged transmit period affect accuracy? In our swarm ranging protocol, a message is transmitted by period, therefore the reply delay (duration between the receive and transmit timestamp) is 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 robot or device (Q3 and Q4)? For simplicity, we define the *ranging period* P to be the time period between two consecutive ranging message transmission. It is clear that the short ranging period results in high frequency ranging[1], which is helpful for fast moving robots and devices. 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 robots or devices are far apart or they move quite slow, the ranging frequency should be set low.

The *ranging period* P should be adapted to velocity and distance. We investigate such adaption in a high velocity but large ranging period scenario. As in Fig. 6, 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, respectively. 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 period, there must be a ranging message received from the other side. Let the receive time divides a transmission period into a $\beta : \alpha$ ratio, where $\alpha + \beta = 1$. Notations and their relationships are detailed in Fig. 6.

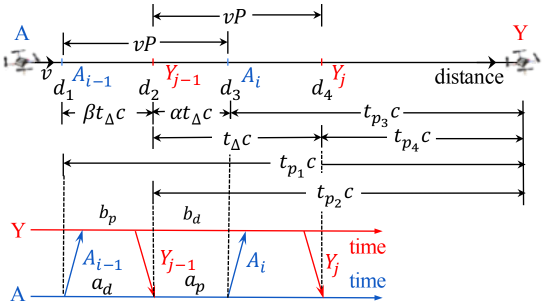


Fig. 6: A high velocity but large ranging period scenario.

Hence, we have $t_{p_1} = t_{p_2} + \beta t_\triangle$, $t_{p_3} = t_{p_2} - \alpha t_\triangle$, and $t_{p_4} = t_{p_2} - t_\triangle$. Therefore, $a_d = b_p + 2t_{p_2} + \beta t_\triangle$ and $b_d = a_p + 2t_{p_2} - \alpha t_\triangle$. 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 following computed ToF

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

note that the last approximation is because $a_p, b_p >> t_{p_2}, t_\triangle$. Because the calculation occurs at receiving Y_j, the actual ToF

$$t_p^{actual} = t_{p_4} = t_{p_2} - t_\triangle = t_p^{computed} - \frac{vP}{c}. \tag{4}$$

If we bound the error by

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

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

$$P \leq \frac{e_0}{1 - e_0} \frac{d_2}{v}. \tag{5}$$

Eq. (5) serves as a guideline to determine the *ranging period* P with given velocity v and computed distance d_2. We can see from this equation that the smaller distance, the shorter ranging period; meanwhile, the faster velocity, the shorter ranging period, which answers Q3 and Q4.

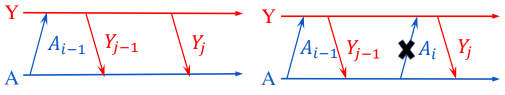
D. Handle ranging period mismatch and packet loss

We have known that, for robot or device A, its ranging period may differ from neighbor to neighbor, according to Eq. (5). We therefore choose the minimum one as the ranging period of A,

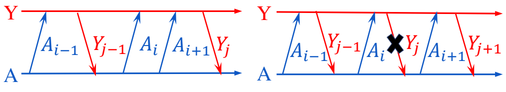
$$P_A = \min_{Y \in \text{neighbors}} \{ \frac{e_0}{1 - e_0} \frac{d_{AY}}{v_{AY}} \}, \tag{6}$$

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

A direct question follows: what if the ranging counterparts have a different ranging period or what if packets are lost in transmission (Q5)? Both questions concern the unbalanced message exchange, which we summarize as four cases in Fig. 7. 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, and we investigate them one by one.



(a) Case 1. (b) Case 2.



(c) Case 3. (d) Case 4.

Fig. 7: The ranging period mismatch and packet loss cases.

---

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

In both Case 1 and Case 2, message Y_j will not carry any receive 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. Our solution discards the timestamps of A_{j-1} and X_i (if it exists), as shown in the lower part of Table II, so that the next round of ranging can be continued.

TABLE II. Handle the mismatch and 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 $T_f =$	$R_e = R_{Y_j}$

⇓

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

Case 3 is easy to handle. We always update/overwrite the transmit 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 transmit 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 is updated into the ranging table such that $T_r = T_{A_j}$. 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 yellow cells of Table III. Our solution discards 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.

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	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_{j-1}}$	$T_f = T_{A_{i+1}}$	$R_e = R_{Y_{j+1}}$

⇓

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

We summarize these steps by `UpdateM`.

Procedure 4: `UpdateM`(tables, case, msg)

```
1 UpdateS(tables, case, $msg$);
2 if $case$==‘rx’ and $msg$ is from neighbor $Y$ then
3     if $R_f == \varnothing$ in table(AY) then
4         $R_r \leftarrow R_e, T_r \leftarrow \varnothing, T_f \leftarrow \varnothing, R_e \leftarrow \varnothing$ for table(AY);
5     else if index($T_r$) $\neq$ index($R_r$) in table(AY) then
6         $R_p \leftarrow R_f, T_p \leftarrow T_f, R_r \leftarrow R_e$ for table(AY);
7         $T_r \leftarrow \varnothing, R_f \leftarrow \varnothing, T_f \leftarrow \varnothing, R_e \leftarrow \varnothing$ for table(AY);
8     end
9 end
```

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.

E. Handle dense and dynamic neighbors

Although the above designed ranging protocol already works smoothly for any small size group, it encounters scalability problem when the number grows. Let us sense the difficulty from the following example. Assume A, B and C are stationary and far apart (but within transmission range), while each of them has a very close and fast moving neighbor. Therefore, A, B and C all have a very short ranging period according to Eq. (6). As a result, the stationary and far apart A, B, C range with each other at a very high frequency, which is unnecessary. This situation becomes even severe when the number and mobility grows. Ultimately, the ranging message capacity will not be enough to carry all timestamps for every neighbor. It is critical to answer Q6: how to select timestamps to fit the message capacity so as to be scalable to dense neighbors?

The high level basic idea of our solution lays in three aspects. (1). For any robot or device, we allow it to have a different ranging period for different neighbors, instead of setting a very small period for all neighbors. So, not all neighbors' timestamps are required to be carried in every ranging message, *e.g.,* the receive timestamp to a far apart and motionless neighbor is required less often. (2). Since different neighbors have different ranging periods, thus the ranging message carries neighbors' timestamps at different frequency. To allow every neighbor gets its turn to have the timestamp carried within the required period, the core idea is maintain the *next (expected) delivery time* for every neighbor. The ranging message capacity is shared according to the chronological order of the *next (expected) delivery time* by all neighbors. (3). When a neighbor is not heard for a certain duration, we set it as expired and will no longer consider it as a neighbor.

We start the detailed design with an improved 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, denoting the newest ranging period for A and Y, the *next (expected) delivery time* for neighbor Y, the expiration time for neighbor Y, respectively. Note that for each neighbor Y, A maintains a ranging table(AY).

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

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

Procedure 5: `Compute`(Table(AY))

```
1 $ToF$ =ComputeS(Table(AY));
2 Update $P$ by Eq. (5) for table(AY);
3 return $ToF$
```

The *next (expected) delivery time* t_n is updated once a message for neighbor Y is delivered, *i.e.,* t_n should be one period 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 `Update`.

With the ranging table improved and appropriately maintained, we now focus on how to generate a ranging message

Procedure 6: `Update`(tables, case, *msg*)

```
`UpdateM`(tables, case, $msg$);
if $case$==‘tx’ then
    for each $Y$ whose receive timestamp is carried in $msg$ do
        $t_n \leftarrow t_{current} + P$ for table(AY);
    end
else if $case$==‘rx’ and $msg$ is from neighbor $Y$ then
    $t_s \leftarrow t_{current} + T_{expiration}$ for table(AY);
end
```

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 receive timestamps. How to choose these m timestamps are critical to generate the ranging message. 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 receive timestamps for these m neighbors is generated and transmitted. Then, the m next delivery times are updated according to their corresponding ranging period by `Update`. The procedure repeats and the next *ranging* message is generated at the nearest *next (expected) delivery time*. Procedure `Generate` presents the details.

Procedure 7: `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;
$RxM \leftarrow \emptyset$;
for each neighbor $Y$ from top $m$ tables do
    $RxM \leftarrow RxM \cup (Y, R_Y)$
end
return Message$(X_i, T_{X_{i-1}}, RxM, v)$
```

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 search for given targets, a swarm of small robots explore and map an unknown indoor environment, an army of legged robot dogs battle 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. 8.

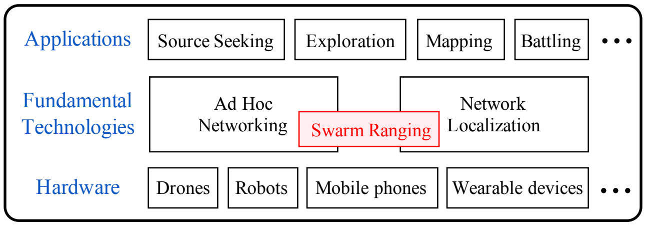


Fig. 8: Swarm ranging supports ad hoc networking and network localization that are fundamental for any swarm applications.

Swarm ranging supports ad hoc networking and network localization simultaneously. (1) In ad hoc networking, 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 [13], 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 [14]. Our swarm ranging protocol provides the fundamental distance information to support higher level localization algorithms. Hereby, Q7 is answered.

IV. IMPLEMENTATION

The proposed swarm ranging protocol is implemented on Crazyflie 2.1 drones, STM32 microcontrollers powered micro drones with 192KB memory and onboard UWB wireless transceiver chips DW1000. 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. 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 involving distance ground truth, an HTC Lighthouse 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 position data.

V. EXPERIMENT

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

To minimize 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 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. 9. It is clear to conclude that the ranging period P do not affect ranging accuracy at various distance.

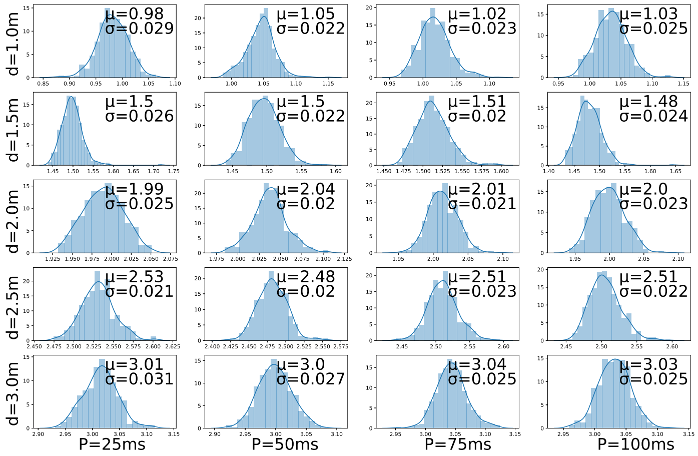


Fig. 9: The impact of ranging period on ranging accuracy.

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 $\bar{P}$=50ms (p=30ms and W=40ms). 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 around 90% and the average *ranging ratio* is around 60%. 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. 7, and thus reduces the *ranging ratio*.

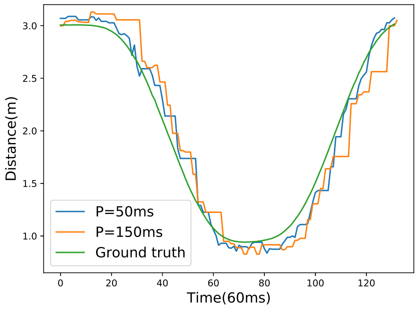
TABLE V. Message reception ratio and ranging ratio.

Ranging Pairs	Receive Count	Reception Ratio	Ranging Count	Ranging Ratio
AB	5451	90.85%	3563	59.38%
AC	5371	89.52%	3540	59.00%
AD	5348	90.53%	3701	61.68%

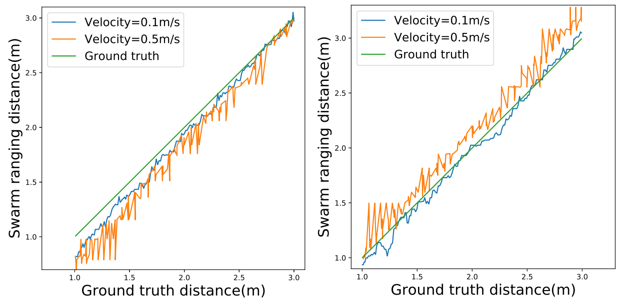
C. *Velocity and ranging accuracy*

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. (4), 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. 10: Ranging accuracy with different velocity and ranging period.

First, the impact of the ranging period is evaluated by two drones A and B. Drone A hovers at a fixed location while drone B flies towards it, starting from a location 3 meters away, both at the same height, 100 cm. B stops at 1 meter distance from A, and then flies back to its starting location. The flight speed for B is set to v=0.5m/s. We repeat twice with the ranging period P= 50ms and 150ms respectively. The ranging results are shown in Fig. 10(a). It is clear that when P becomes larger, the accuracy drops. We also observe that the ranging results are delayed compare to the ground true, and the larger P, the bigger delay, which is consistent with our theoretical analysis in Eq. (4).

Second, the impact of flight speed is investigated by the same settings. The differences are we now fix the ranging period to $\bar{P}$=50ms (p=45ms and W=10ms), 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. 10(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 speed 0.5m/s. As can be seen in Fig. 10(b)(c), the ranging results are more stable for v=0.1m/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 $\bar{P}$ = 30ms, 40ms, 50ms, and 60ms for A, B, C, and D, respectively, where the random window size is set W = 40ms for all drones. Fig. 11 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 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.

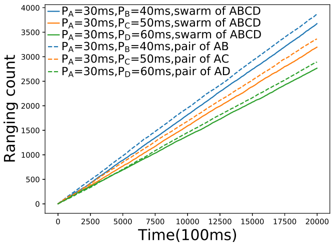


Fig. 11: Performance for mismatched ranging period.

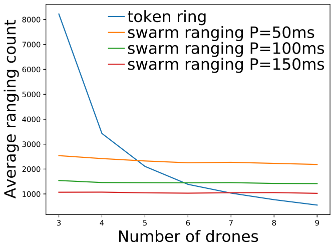


Fig. 12: Comparison with ranging based on token ring.

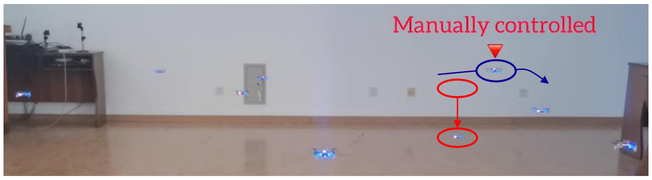


Fig. 13: Crazyflie drones avoiding collision when flying compactly.

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 = 80ms. We vary the numbers of drones, from 3 to 9. Fig. 12 shows the ranging counts recorded by drone A for 200s.

Any point in Fig. 12 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. 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=hJ8yo2ReBdA.

In this experiment, 8 Crazyflie drones hover at height 70cm in a compact area less than 3m by 3m. While a ninth Crazyflie drone is manually controlled to fly into this area. We set the average ranging period $\bar{P}$=100ms (p=60ms and W=80ms) and the maximum received capacity m=8. As shown in the demo video and Fig. 13, thanks to the swarm ranging protocol, a drone detects the coming drone by ranging distance, and lower its height to avoid collision once the distance is small than a threshold, 30cm.

VI. RELATED WORK

UWB are being used widely for indoor localizations [15], [16], [17], [18]. Tiemann *et al.* [15] and Corbalan *et al.* [16] design TDOA based algorithms for ranging which require time synchronization. Xu *et al.* [17] fuse visual, inertial, and UWB information for aerial swarm localization. Cao *et al.* [18] locates a robot with only a single UWB anchor (base station).

Recently, concurrent ranging is becoming an emerging trend in ultra-wideband research community [19], [20], [21], [22]. Corbalan *et al.* [19], [20] 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.* [22] later extends the idea to reflection resilient. While Stocker *et al.* [21] extends it to support ranging for unlimited number of tags.

There are also a few work focus on UWB ranging for large numbers [16], [21] or for high mobility [23]. Corbalan *et al.* [16] and Stocker *et al.* [21] focus to locate countless tag by TDOA or by concurrent ranging. Their work is dedicated for the anchor-tag model, not applicable for swarm scenario. Risset *et al.* [23] investigate the UWB ranging problem for rapid movements, with only two UWB tags.

UWB is also been used for data communication [24], [25]. Mohammadmoradi *et al.* [24] propose to simultaneous ranging and communication in UWB networks. Vecchia *et al.* [25] investigate the concurrent transmission problem for UWB.

However, none of the above related work focus on a ranging protocol design specially for dynamic and dense swarms.

VII. CONCLUSION

This paper proposes a UWB swarm ranging protocol, designed specially for dynamic and dense robotic or device swarms. 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 packet loss is handled appropriately; it is *scalable and compatible* because a rotation scheme is designed to handle dense neighbors and higher level networking and localization protocols and algorithms are supported. Finally, this protocol is implemented on Crazyflie drones, that are powered by STM32 microcontrollers and have only 192KB memory. Thanks to the swarm ranging protocol, a total of 9 Crazyflie drones can automatically avoid collision when flying in a compact space.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China Grant 2018AAA0101200, 2018YFB2100300, National Natural Science Foundation of China Grants 62072101, 61632008, 61972086, 62072102, 61872079, and 62072099, Jiangsu Provincial Key Laboratory of Network and Information Security Grant BM2003201, and Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China Grant 93K-9, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

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

[2] "RoboMaster EP - DJI," https://www.dji.com/robomaster-ep, accessed Jul. 30, 2020.

[3] "Boston Dynamics Launches Commercial Sales of SPOT robot," https://www.bostondynamics.com/press-release-spot-commercial-launch, accessed Jul. 30, 2020.

[4] 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 Communications*, vol. 10, no. 1, pp. 1–12, 2019.

[5] Y. Yang and M. A. Bevan, "Cargo capture and transport by colloidal swarms," *Science Advances*, vol. 6, no. 4, p. eaay7679, 2020.

[6] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, p. eaaw9710, 2019.

[7] 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, 2019.

[8] Decawave, *DW1000 user manual*. Decawave Ltd, 2017.

[9] "iPhone11 - Apple," https://www.apple.com/iphone-11/specs, accessed Aug. 11, 2020.

[10] "IEEE Standard for Local and metropolitan area networks - Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std. 802.15.4-2011*, pp. i–294, 2011.

[11] D. Neirynck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method," *Proc. 2016 13th Work. Positioning, Navig. Commun. WPNC 2016*, pp. 16–19, 2017.

[12] B. Broecker, K. Tuyls, and J. Butterworth, "Distance-Based Multi-Robot Coordination on Pocket Drones," in *2018 IEEE Int. Conf. Robot. Autom.* IEEE, may 2018, pp. 6389–6394. [Online]. Available: https://ieeexplore.ieee.org/document/8461176/

[13] "Optimized Link State Routing Protocol (OLSR)," Internet Requests for Comments, RFC Editor, RFC 3626, 10 2003. [Online]. Available: https://tools.ietf.org/html/rfc3626

[14] M. Z. Win, W. Dai, Y. Shen, G. Chrisikos, and H. Vincent Poor, "Network Operation Strategies for Efficient Localization and Navigation," *Proc. IEEE*, vol. 106, no. 7, pp. 1224–1254, 2018.

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

[16] P. Corbalán, G. P. Picco, and S. Palipana, "Chorus: UWB concurrent transmissions for GPS-like passive localization of countless targets," in *Proc. IPSN*, 2019, pp. 133–144.

[17] 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. ICRA*, 2020, pp. 8776–8782.

[18] Y. Cao, C. Yang, R. Li, A. Knoll, and G. Beltrame, "Accurate position tracking with a single UWB anchor," in *Proc. ICRA*, 2020.

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

[20] P. Corbalán and G. P. Picco, "Ultra-wideband Concurrent Ranging," *ACM Trans. Sens. Networks*, 2020. [Online]. Available: http://arxiv.org/abs/2004.06324

[21] 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," in *Proc. IPSN*, 2019, pp. 348–349.

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

[23] T. Risset, C. Goursaud, X. Brun, K. Marquet, and F. Meyer, "UWB Ranging for Rapid Movements," in *IPIN 2018 - 9th Int. Conf. Indoor Position. Indoor Navig.* IEEE, 2018.

[24] H. Mohammadmoradi, M. Heydariaan, and O. Gnawali, "SRAC: Simultaneous ranging and communication in UWB networks," *Proc. - 15th Annu. Int. Conf. Distrib. Comput. Sens. Syst. DCOSS 2019*, pp. 9–16, 2019.

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