

# 超宽带群测距协议

## 动态和密集网络

单峰, IEEE 会员, 霍浩东, IEEE 研究生会员, 曾嘉欣, 李增宝,  
Wu Weiwei, IEEE 会员, Luo Junzhou, IEEE 会员, ACM 会员

**摘要:**如今, 不仅是可穿戴和便携式设备, 空中和地面机器人也可以被制造得更小、更轻、更便宜, 因此数百个机器人可以形成一个群体, 参与复杂的合作应用, 如搜索、救援、地图绘制和作战。蜂群中的设备和机器人具有数量多、机动性高、距离短三个重要特点, 形成动态密集的无线网络。成功的群合作应用需要低延迟通信和实时本地化。本文提出使用超宽带(UWB)无线电技术来实现这两种功能, 因为 UWB 对时间非常敏感, 可以使用数据消息的发送和接收时间戳来计算准确的距离。设计了一种超宽带群测距协议, 可以同时实现无线数据通信和群测距, 使设备/机器人可以同时计算到所有邻居的距离。该协议适用于动态、密集的网络, 同时也可用于各种无线网络, 并在各种类型的设备/机器人(包括低端设备)上实现。在我们的实验中, 该协议被实现在 Crazyflies, STM32 微控制器驱动的微型无人机上, 机载超宽带无线收发芯片 DW1000。在一个紧凑的区域内, 共有 9 架 crazyfly 无人机进行了大量的真实世界实验, 以验证所提出的协议在各个性能方面的有效性。实现的蜂群测距协议是开源的, 网址是 <https://github.com/SEU-NetSI/crazyfly-firmware>

**索引术语:**超宽带, 蜂群, 测距, 协议设计, 网络, 双向测距, 疯狂飞行。

稿件于 2021 年 7 月 24 日收到;2022 年 5 月 4 日修订;2022 年 6 月 11 日受理;由 IEEE/ACM 网络学报编辑 R. Lo Cigno 批准。本研究部分获得国家重点研发计划(2018AAA0101200)资助;部分获国家自然科学基金(62072101、61972086、62072102、62172091、62132009)资助;部分项目获江苏省网络与信息安全重点实验室 BM2003201 资助;教育部计算机网络与信息集成重点实验室 93K-9 项目;部分由浙江实验室资助, 授权 2019NB0AB05。该论文的初步版本在 2021 年 IEEE 计算机通信国际会议(IEEE INFOCOM 2021)上部分展示[DOI: 10.1109/INFOCOM42981.2021.9488717]。(通讯作者: 冯山;Junzhou 罗)。

来自东南大学计算机科学与工程学院的单峰、吴薇薇、罗俊洲(e-mail: shanfeng@seu.edu.cn;weiweiwu@seu.edu.cn;jluo@seu.edu.cn)。

霍浩东、李增宝来自东南大学网络科学与工程学院, 江苏南京 210096 (e-mail: huohaodong@seu.edu.cn;lizengbao@seu.edu.cn)。

曾嘉欣就职于中国银联股份有限公司技术开发中心, 上海 200135 (e-mail: zengjiaxin@unionpay.com)

数字对象标识 10.1109/TNET.2022.3186071

我的介绍。

随着电子制造业的快速发展, 越来越多的可穿戴和便携

设备、空中和地面机器人正在商业化。例如, 各厂商的智能手机和智能手表都在不断发布和升级, 比特热在 2019 年 2 月发布了微型无人机 crazyfly 2.1[1], 大疆在 2020 年 3 月发布了 RoboMaster EP[2], 波士顿动力在 2020 年 6 月推出了 SPOT 腿机器人[3]。

如今, 不仅可穿戴和便携式设备, 空中和地面机器人也可以被制造得体积更小、重量更轻、价格更低, 因此它们在许多领域得到越来越广泛的应用。现在可以利用多达数百个的蜂群来参与复杂的合作应用。以下是一些例子, 一组室内无人搜索给定的目标, 一群小型机器人探索并绘制未知的室内环境, 一群有腿的机器狗在森林深处战斗。与功能齐全的大型单个机器人相比, 一群小型设备和机器人具有容错更高、部署规模和数量更灵活、部署速度更快的优点。因此, 对设备和机器人群的研究是当前的趋势, 在 Nature 和 Science[4] - [7]等顶级期刊上出现了文章。

我们注意到群体中的设备和机器人有三个重要特征, 即数量多、机动性高、距离短。首先, 根据任务的复杂性, 可能会部署数百个设备和机器人进行协作。第二, 人类携带的可穿戴和便携式设备, 微型无人机, 轮式和腿式机器人, 都可以快速响应请求并做出动作。第三, 由于体积小, 他们可以在短距离内合作, 以完成复杂的任务。因此, 在未来的应用中, 蜂群中的设备和机器人应该是动态的、密集的, 从而形成动态的、密集的无线网络进行协作和通信。

成功的群合作应用需要低延迟通信和实时本地化。特别是在没有外部支持基础设施的情况下, 群内设备与机器人之间的自组网和相对定位尤为重要。因此, 一个高效的设备间/机器人通信协议, 支持

本作品根据知识共享署名-非商业衍生品 4.0 许可协议授权。要了解更多信  
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

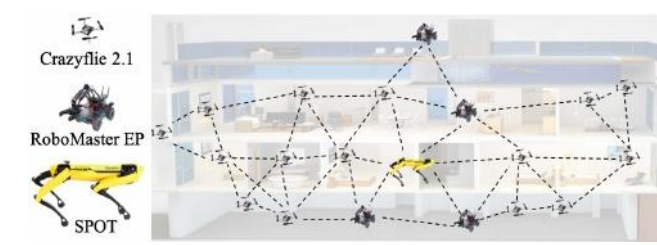


图 1 所示。设备群和机器人是动态的、密集的。

同时传送数据和遥感距离(即是基本的基础)。

本文提出利用超宽带(UWB)无线电技术实现蜂群的组网和测距。因为 UWB 对时间非常敏感，它为任何一对设备之间交换的每一个数据消息提供了高度准确的传输时间戳和接收时间戳，通过它可以在数据通信过程中不断计算和更新每一对设备之间的实时距离。在这里，事件的时间戳(传输或接收)是事件发生的时间时刻。

因此，本文重点设计了一种 UWB 群测距协议，该协议可以同时实现无线数据通信和群测距，群测距是一种实时机制，允许设备/机器人同时计算到其所有邻居的距离。注意，远程距离是定位和定位的基础。现有的基于超宽带的测距方法没有考虑动态和密集的网络，这与我们的方法不同。

我们现在将主要挑战总结如下。

**大量。**一个蜂群可能包含大量的设备和机器人，它们必须访问一个共享的公共无线信道。由于无线信道的广播特性，不适当的协议可能会导致频繁的传输冲突，从而延长成功的测距操作，降低其频率。此外，由于设备数量较多，协议必须尽量减少测距作业所需的消息数量，并尽量减少低端设备和机器人的计算时间。因此，如何设计一种简单高效的超宽带群测距协议是一项具有挑战性的任务。

**高流动性。**通常，需要高频率的测距操作来实现高机动性。然而，有时设备和机器人也可能需要保持静止，因为应用程序需要。在这种情况下，为了节省无线信道资源，需要降低测距频率。此外，高移动性会导致无线通信不稳定，导致测距信息丢失。因此，如何设计一种自适应的、鲁棒的 UWB 群测距协议来解决高移动性问题是另一项具有挑战性的任务。

**短的距离。**在一个群体中，两个设备/机器人之间的距离可能很短，因此一个设备或机器人可能有许多邻近的邻居来进行测距。但是，距离消息有大小限制，不能携带距离信息(时间戳)供所有邻居计算距离

一次。此外，一个好的测距协议应该支持和兼容现有的网络和本地化协议。因此，如何设计一种可扩展且兼容的超宽带群测距协议也是我们需要应对的挑战。

总之，本文的目标是为动态和密集网络设计一种超宽带群测距协议，有效地计算每个设备或机器人与所有邻居的距离。本文的贡献总结如下。

- 1)我们设计的协议似乎是一项开创性的工作，利用超宽带技术在动态密集网络中同时实现无线测距和数据通信，具有广泛的应用前景，可用于各种无线网络，包括机器人网络、体域网络等。
- 2)我们的协议高效且易于实施。这是非常有效的，因为我们的协议设计利用了无线通信的广播特性。它很容易在所有类型的设备/机器人(包括低端设备)上实现，因为它非常简单，只需要一种消息类型。
- 3)我们的测距协议可以平滑地处理动态网络，因为它可以根据每对中两个设备之间的距离变化，实时自适应地调整测距频率。该协议还可以处理消息丢失。
- 4)我们的测距协议平滑地处理密集的网络。设计了一种基于公交上车的方案，用于处理设备/机器人的邻居太多，限制大小的测距消息无法携带所有邻居的测距信息(时间戳)来一次性计算距离的情况。
- 5)我们用 STM32 微控制器驱动的微型无人机 Crazyflies 实现了该协议，内存只有 192KB，机载 UWB 无线收发芯片为 DW1000。在一个紧凑的区域内，共有 9 架 crazyfly 无人机进行了广泛的真实世界实验，以评估拟议协议的各种性能方面。我们的实现是开源的，网址是 <https://github.com/SEU-NetSI/crazyflie-firmware>。

本文的其余部分组织如下。第二部分介绍了超宽带和测距协议的基础知识。第三部分给出了群测距协议的详细设计。第四节给出了实现细节。第五节讨论了实验结果，第六节总结了相关工作，第七节总结了本文，讨论了未来可能的改进和实际实施工作。参考文献[8]报告了我们的初步工作。

2 初步

A.超宽带(UWB)无线电技术

超宽带(UWB)是一种在宽频谱上以高比特率传播数据的无线传输技术。超宽带技术的一个特点是使用宽度为 1ns 的薄脉冲来传输比特。之间的差异

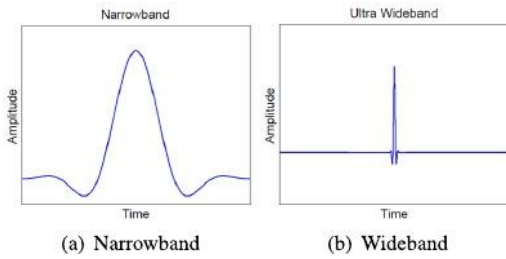


图 2 所示。与常见的窄带无线电技术相比，超宽带技术使用短而尖锐的薄脉冲传播数据，足以提供准确的时间戳。

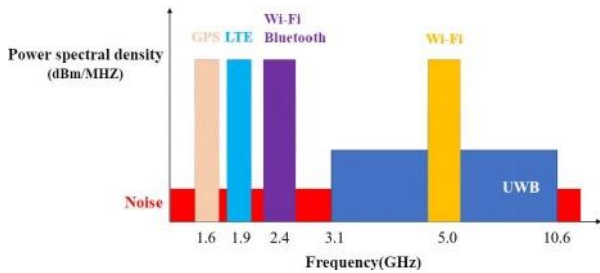


图 3 所示。其宽频带和低谱密度使设备能够以低能耗传输数据。

常见的窄带无线电技术(如 Wi-Fi、LTE)和超宽带技术如图 2 所示。给定脉冲的大小和形状，就可以非常精确地测量飞行时间[9]。

典型的超宽带无线电收发芯片的无线电频率范围为 3.5 GHz ~ 6.5 GHz，测距精度小于 10 cm[10]。如图 3 所示，UWB 的宽频带和较低的谱密度使得 UWB 设备能够以较低的能耗传输数据。超宽带具有较强的抗多径能力和突防能力，在复杂环境下也具有良好的性能。

利用超宽带在定位和数据传输方面的突出特点，电信公司正在逐步布局超宽带市场。苹果公司已经将其自主研发的超宽带芯片(U1)应用到消费者级产品中，比如最新的 iPhone 和 Apple Watch。三星、小米等企业也提出了超宽带解决方案。三星和苹果最近都在各自的产品生态系统中推出了超宽带便携式追踪设备 SmartTag Plus[11]和 AirTag[12]。由于其优越的安全性和较高的测距精度，超宽带无线电技术现在正逐渐被应用到越来越多的应用中，如汽车钥匙、动作捕捉、物联网设备定位和消息交互等。最新的 IEEE 标准 802.15.4z-2020[13]中定义了超宽带精密测距规范。可以预见，未来超宽带技术将在越来越多的场景中得到应用。

### B. 双向测距(DS-TWR)协议

IEEE 标准 802.15.4z-2020[13]中定义了一个全新的标准化测距协议，即双面测距

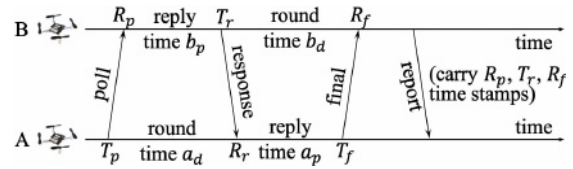


图 4 所示。双向测距(DS-TWR)协议。

本文简要介绍一种双向测距(DS-TWR)协议。

如图 4 所示，有四种类型的消息，即，*poll*, *response*, *final*, *report* 消息，按顺序在双方 A 和 B 之间交换，用于每次测距操作。前 3 条消息的发送时间戳和接收时间戳分别表示为  $T_p$ 、 $R_p$ 、 $T_r$ 、 $R_r$ 、 $T_f$ 、 $R_f$ 。每一方从一个接收时间到下一个发送时间的持续时间称为应答时间，从一个发送时间到下一个接收时间的持续时间称为一轮时间。让我们在图 4 中表示双方的这些持续时间如下。

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

$t_p$  为飞行时间(ToF)，即无线电信号从一侧到另一侧的传播时间。那么，两轮时间持续时间可以写成  $a_d t = 2p + b_p b_d = 2t_p + a_p$ ，因此，ToF 可以计算为[10], [13]为

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

(2)式由于包含 4 个参数，其误差控制效果优于  $t_p = (a_d - b_p)/2$  等公式。由于每边都有一个由其晶体偏移引起的时钟偏移，因此 ToF 的实际计算值可以表示为，其中  $e_A$  和  $e_B$  分别为 A 边和 B 边的时钟误差。

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

其与式(2)的真实值的偏差为

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

可以看出，当  $e_A$  和  $e_B$  很小时，偏差也很小。

值得注意的是，在大多数设备和机器人的操作系统中，传输时间戳只有在消息发送后才可用。因此，需要一个附加的报告消息来携带三个相关的时间戳，以便进行 ToF 计算。

### 基于令牌环的 DS-TWR 算法

对 DS-TWR 协议提出了一个简单的扩展[14], [15]来处理多对多的测距操作，该协议使用令牌环技术来控制测距过程。基本的想法是所有相邻的设备形成



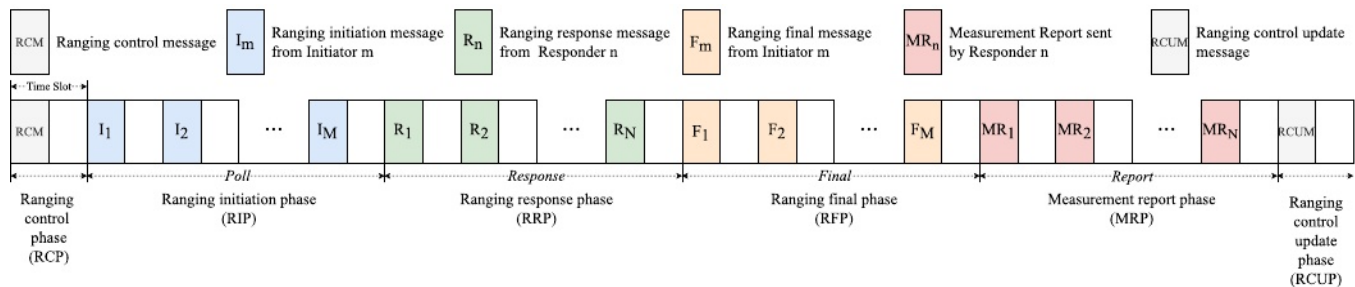


图 5 所示。新修订的标准将 DS-TWR 协议扩展到多对多场景，在一个测距回合中有六个测距阶段。

一枚戒指，只有一个代币。因此，在任何时候只有一个设备持有令牌。token 持有者依次为每个邻居发起 DS-TWR 协议。一旦令牌持有者完成了所有邻居的范围，令牌将被传递给环中的下一个设备。

虽然这个扩展很容易理解，但它有一些缺点。(1)每当双方交换消息时，由于无线通信的广播性质，这些消息可以被许多邻居听到。然而，它们被忽略，这使得协议效率低下。(2)为了形成令牌环，所有的邻居必须事先知道彼此。这就增加了一个额外的邻居发现和失效机制，使得协议的可扩展性很差。(3)由于无线通信不可靠，消息可能会丢失。当令牌丢失时，整个范围进程将停止。因此，token 必须被监控并从损失中恢复，这将使其实现复杂化。

#### D. IEEE 802.15.4z-2020 中 DS-TWR 的扩展

新修订的 IEEE 标准 802.15.4z-2020 还包括基于 DS-TWR 协议的多对多测距协议。该协议以距离为基本时间单位，每轮距离更新一次。更具体地说，每一轮测距包括 6 个测距阶段，其中 4 个阶段专门用于发送投票、响应、最终和报告消息。每个阶段的时间进一步开槽和有足够的位置在每个阶段每个设备收集足够的时间戳从所有邻国计算根据情商 ToF。(2)给出更多的细节。在图 5 中,我们可以得出新的协议是一种改进 DS-TWR 因为这四个类型的消息扮演类似的角色如 DS-TWR 协议,但每个需要更多的时间槽,因为启动装置等可能会有很多的邻居。

虽然这个来自新的 IEEE 标准的多对多范围协议是 DS-TWR 的一个很好的扩展，但它缺乏对动态和密集网络的支持。首先，媒体访问方法是基于预先分配的时隙，这对于动态连接和离开邻居是不友好的。由于每隔一轮测距就更新一次时隙分配，新的邻居必须等待下一轮测距才能加入测距操作。其次，对于密集的网络，一轮测距可能需要很长时间才能完成，因为

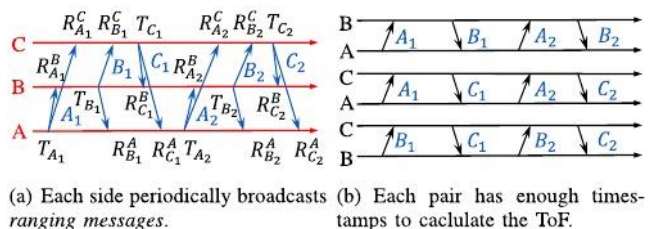


图 6 所示。新协议的说明，为三方面的例子。

设备或机器人可能有许多邻居来进行测距，每个邻居必须被分配多个时隙。因此，网络越密集，测距周期就越长。

上述对超宽带测距协议现状的简要调查和分析，启发我们设计一种新的群测距协议，该协议将改进目前已知的协议。

#### E. 一种新的群体测距协议的基本思想

回想一下，DS-TWR 协议有四种类型的消息在双方之间按顺序交换。其中一方发起投票消息后，双方轮流回复回复消息、最终消息和报告消息。对于新的蜂群测距协议，我们只使用了一种消息类型，即测距消息。我们让每一方周期性地传输一个不响应任何特定传入消息的范围消息。新协议仍将根据 Eq.(1)和(2)使用 6 个时间戳来计算 ToF。在下面的例子中，我们将使用一个三面玩具的例子来解释如何仅使用一种类型的消息来完成范围操作。这个示例有助于理解新协议背后的内在逻辑。

图 6 显示了在执行新协议期间，玩具示例中每一方传输的测距消息的序列。

如图 6 所示，A、B、C 三面轮流传递 A1、B1、C1、A2、B2、C2 六个消息。由于无线通信的广播性质，每条消息都可以被其他双方接收。然后每条消息生成三个时间戳，即，一个传输时间戳和两个接收时间戳，如图 6(a)所示。我们可以看到，如图 6(b)所示，每对都有两轮消息交换。因此，有足够的时间戳来计算每对的 ToF。

每边只传输两条消息，就可以计算出所有三对的距离，这比 IEEE 802.15.4z-2020 标准中基于令牌环的扩展和许多 DS-TWR 扩展高效得多。这一发现启发我们设计一种新的群测距协议。但是，对于动态密集的设备 and 机器人网络，我们必须应对数量大、机动性高、距离短的挑战。总之，必须回答以下问题。

- Q1(简单但有效)我们应该如何设计范围消息的格式，以便有足够的时间戳来计算 ToF?
- Q2(简单而有效)放大的传输周期如何影响精度?我们需要回答这个问题，因为在 DS-TWR 中，消息是立即回复的，但在新协议中，消息是定期发送的。
- Q3(自适应)高机动性如何影响测距精度?如果测距信息的传输频率一直很低，但是设备或机器人的移动速度变快了，我们需要知道测距精度会受到怎样的影响?
- 问 4(自适应)测距讯息应多久广播一次?当两个设备或机器人距离较远或移动速度较慢时，测距频率可以较低，但当它们距离较近或移动速度较快时，则必须较高，以免发生碰撞。但是标准是什么呢?
- Q5(鲁棒)如果消息丢失或测距频率不匹配导致双方消息交换不平衡，我们该怎么办?
- Q6(可扩展)当一个设备有很多邻居时，我们应该如何处理密集的蜂群?我们需要回答这个问题，因为在这种情况下，一个范围消息没有空间来携带所有的时间戳。
- Q7(兼容)新协议是否支持和兼容更高级别的协议，如优化的链路状态路由协议和网络定位算法?

3 一种新的蜂群测距协议

A. 测距消息及主框架的设计

为了符号的一致性和清晰性，让  $A_k, k = 1, 2, \dots$ ，表示设备 A 传输的第  $k$  条消息，让  $TA_k$  表示  $A_k$  的传输时间， $RAY$  表示时间  $k$  消息  $A_k$  被 Y 接收，我们有时使用简单的符号  $RA_k$  如果没有歧义。

回想一下图 6 中的三面示例，让我们将重点放在定位消息 C2 上。在接收到 C2 后，A 和 B 应该能够分别计算到 C 的 ToF，因为根据 Eq.(1)和(2)，他们各自收集了这个 ToF 计算所需的 6 个(不同的)时间戳。更具体地说，设备 A 收集了  $TA_1, RAC, TC_1, RCA, TA_2, RAC$ 。类似地，设备 B 已经收集到  $TB_1, RB_1, TC_1, RC_1, TB_2, RCB$ 。因此 C2 必须包含 C1 的传输时间戳(只有在 C1 传输完成后才可用)，以及 A2 和 B2 的接收时间戳。

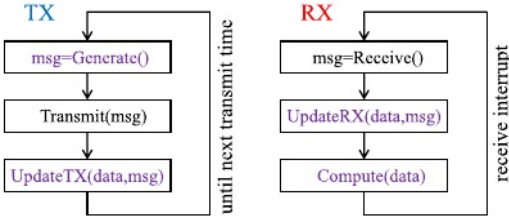


图 7 所示。群测距协议的主要框架:传输(TX)部分和接收(RX)部分。

定义为

$TX_{i-1}$  以前的消息  $X$  的传输时间戳是  $t_{i-1}$ ;  $M_{ix}$  是邻居发送的消息的接收时间戳集合，例如  $M_{ix} = \{(A_2, RA_2), (B_2, RB_2), \dots\}$ ，这应该足以以为每个邻居计算电流的 ToF;  $v$  是  $X$  的当前速度。

蜂群测距协议的主要框架分为两部分:，发送(TX)部分和接收(RX)部分。两个部分都由每个设备 X 执行，如图 7 所示。TX 部分是偶尔执行的。它包括生成消息、广播消息和更新测距数据 3 个步骤。一旦接收到消息，就会执行 RX 部分。它还包括 3 个步骤，即获取消息、更新测距数据和计算距离。

程序 Generate、UpdateTX、UpdateRX、Compute 和测距数据是群测距协议设计的关键。

B. 简单场景中的消息生成和数据更新

让我们从一个简单的场景开始协议设计，在这个场景中，有许多设备或机器人，a、B、C 等，在很短的距离内。它们中的每一个都会周期性地传输所有其他人都能听到的测距信息，并且它们以相同的速度广播信息。因此，在任何两个连续的消息传输之间，一个设备或机器人可以听到所有其他设备的消息。范围消息被设计为包括所有这些接收时间戳以及它们的消息标识。伪代码在过程 generate 中给出。

操作步骤 1 生成( $A_i, TA_{i-1}, v$ )

1. 初始化  $M_i^x \leftarrow \emptyset$ ;
2. 对于自上次传输以来收到的每个消息  $Y_j$  所做的
3.  $M_i^x \leftarrow M_i^x \cup (Y_j, R_{Y_j})$
4. 结束
5. 返回消息 ( $A_i, TA_{i-1}, M_i^x, v$ )

让我们关注其中一对 A 和 Y 之间的消息交换过程，其中 Y 可以是 B、C 等。前几个消息交换如图 8 所示，其中

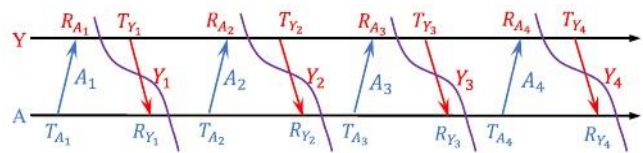


图 8 所示。A 和 Y 之间的最初几个消息交互(可以是 B, C, 等。

表一是测距表数据结构的一个简单例子

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

每个被传递的消息都有一个传输时间戳和一个接收时间戳。从 A 的角度来看，在接收到消息 Y1 时，对 RA 进行了时间戳 i，RY 根据 Y 的生成过程可知。由于消息 Y2 的最后传输时间为 TY1，其最新接收时间 RA；等接收到 Y2 时，有 7 个时间戳可用(紫色第二行左侧)，前 6 个时间戳可用公式 (1)(2)计算 ToF。同理，接收到 Y3 和 Y4 时，可以计算新的 ToF。

通常，只要接收到范围消息，就可以计算出一个新的 ToF，并且计算只需要一小部分时间戳。因此，我们提出了一个范围表数据结构，如表 I，它存储最新的 7 个时间戳。每次接收到一个测距消息，就计算 ToF，并更新测距表。

假设 Ai-1 是最后一个发送的消息，Yj-1 是最后一个从 Y 接收到的消息，那么在范围表数据结构(表 I)中，新一轮的 Tp、Rp 和 Rr 是最初已知的。传递消息 Ai 后，Tf 更新为 TAi。正式的更新步骤在程序 UpdateTX-S 中。在接收 Yj 时，它的接收时间戳 RYj 可以更新为 Re。同时，Yj 带来了两个 Y 侧时间戳，TYj-1 和类风湿性关节炎 i，可以更新 Tr 和 Rf。更新的正式步骤在过程 UpdateRX-S 中。

UpdateTX-S(表, msg)
1.对于每个现有的表中表做
2 更新它的 Tf。
3 结束
UpdateRX-S(表, msg)
1 如果 msg 来自邻居 Y，那么
如果表 (AY) 不存在
3 初始化表 (AY)；
4 结束
5 更新 Tr, Rf 和 Re 表 (AY)；
6 结束

注意，A 可能有许多邻居，因此它需要维护许多范围表，每个邻居对应一个。每一次

当设备或机器人收到一个新的邻居的测距消息时，它会为这个邻居初始化一个新的测距表。对于发送的消息，发送的时间戳要更新到每个范围表中，而对于已经存在的邻居的消息，接收的时间戳要更新到邻居对应的范围表中。

更新测距表后，ToF 的计算方法如计算过程所示。注意，第 3 行和第 4 行中的表更新是为下一轮 ToF 计算做准备。

程序 4 计算(表(AY))

1 如果 Table(AY)不完整，则返回∅；
2 利用表(AY)中的数据，通过公式(1)、(2)计算 ToF；
3Rp ←Rf, Tp ←Tf, Rr ←Re；
4Tr ←∅, Rf ←∅Tf ←∅, Re ←∅；
5 返回 ToF

由此回答了 Q1，即。，被包含在下一个范围消息的时间戳在过程 generate 中，更新范围表的步骤在过程 UpdateTX-S 和 UpdateRX-S 中。

我们现在看 Q2，即。，加大传输周期是否会影响精度？在我们的群排协议中，消息是周期性发送的，因此应答延迟(从接收时间戳到发送时间戳的时间间隔)明显大于立即回复消息的 DS-TWR 协议。但由式(3)可知，计算出的 ToF 的精度仅与两个器件晶体的偏移误差有关，与应答延迟无关。所以，问题得到了答案。

C. 自适应测距协议设计

一些重要的问题仍然是开放的:如何设置一个合适的测距信息传输周期的快速移动设备或机器人(Q3 和 Q4)?我们将测距周期 P 定义为两次连续测距消息传输之间的时间周期。很明显，测距周期短，测距频率高，<sup>1</sup>这对快速移动的设备 and 机器人是有帮助的。但是，较短的测距周期可能会导致无线信道被过多的测距消息占用，从而导致消息冲突，降低同一信道上的数据通信吞吐量。当两个设备或机器人相距较远或移动较慢时，测距频率应设置较低。

测距周期 P 应与速度和距离相适应。我们研究了这种适应在高速但大范围周期的情况下。如图 9 所示，假设 A 以相对速度 v 向 Y 移动，设 P 为 A 和 Y 的测距周期，在此期间，A 移动 vP 距离。假设 A 发送 Ai-1，接收 Yj-1，在 Y 方向的 d1、d2、d3、d4 处依次发送 Ai、接收 Yj。所以 tp1，tp2，tp3 和 tp4 分别是无线信号传输这些距离的持续时间。定义 tΔ 为无线信号传播 vP 距离的时间，因此 tΔc = vP，其中 c 为光速。在一个传输

<sup>1</sup> 本文将测距和 ToF 计算互换使用。



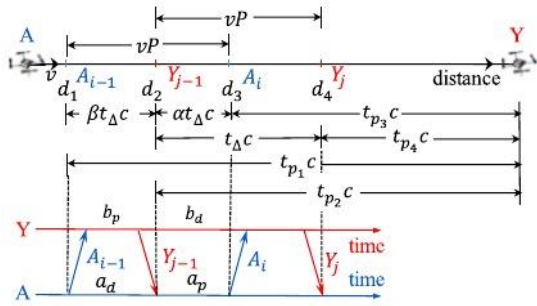


图 9 所示。高速但大范围周期的场景。

期间，必须有从对方接收到的测距消息。让接收时间将一个传输周期划分为  $\beta : \alpha$ ，其中  $\alpha + \beta = 1$ 。图 9 详细介绍了符号及其关系。

因此，有  $t_{p1} = t_{p2} + \Delta \beta t$ ,  $t_{p3} = t_{p2} - \alpha t \Delta$ ,  $t_{p4} = t_{p2} - t \Delta$ . Therefore,  $a_d = b_p + 2t_{p2} + \beta t \Delta$  and  $b_d = \alpha t + 2t_{p2} - t \Delta \alpha$ . 因为处理时间要长得多

比信号传播时间，所以  $a_d \approx b_p$  因此  $b_p : a_p = \beta : \alpha$  因此，我们得到计算的 ToF 如 Eq.(4)所示，如下一页底部所示。

注意最后的近似是因为  $a_p, b_p \gg t_{p2} t \Delta$ 。位置  $d_4$  指向 Y 方向的实际 ToF 应为

$$t_p^{actual} = t_{p4} = t_{p2} - t \Delta = t_p^{computed} - \frac{vP}{c}. \quad (5)$$

如果我们想用常数  $e_0$  来限制误差，

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

然后,我们得  $\frac{t \Delta}{t_{p2}} \leq \frac{1 - e_0}{1 + e_0}$ 。因为  $vP = t \Delta c$ ，我们

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

公式(6)以给定速度  $v$  和计算距离  $d_2$  为准则，确定测距周期  $P$ 。

然而，在实践中，双方都有由晶体偏移引起的时钟偏移。如前所述，设晶体误差分别为  $e_A$  和  $e_Y$ 。因此，实际的测距周期

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

分别。因此助教  $\Delta c = \Delta c_A$ 。如前所述，接收时间除以  $P_A$

转化为  $\beta : \alpha$  的比值，其中  $\alpha + \beta = 1$ 。使用相同的推导步骤，

我们将得到  $tp1 = tp2 + \beta t \Delta$

和  $tp3 = tp2 - \alpha t \Delta$ 。因此  $ad A = bp Y + 2tp2 + \beta t \Delta A$  and  $A$

$b_d^Y = a_p^A + 2t_{p2} - \alpha t \Delta$ . It is worth noting that both sides have

它们自己的时钟偏移量，因此  $ad A = ad(1 + e_A)$ ，该规则也用于  $aAp, bY$  and  $bYp$ 。

因此，考虑晶体误差的影响时，实际计算出的 ToF 为式 (7)，如图下页所示。

Eq.(7)中的最后一个近似是正确的，因为  $a_p, b_p \gg t_{p2}$ ,  $t \Delta$  和  $e_A, e_Y$  相对来说很小。

综上所述，即使引入晶体误差，式(6)仍然可以作为确定测距周期  $P$  的依据。

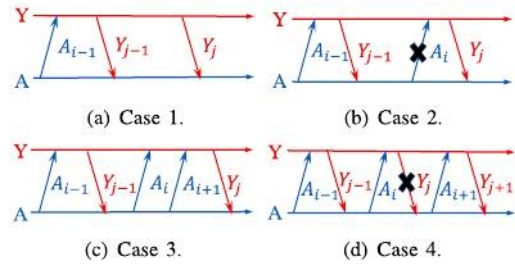


图 10 所示。测距周期不匹配和消息丢失的情况。

最后由式(6)可知，距离越小，测距周期越短；同时，速度越快，测距周期越短，解决了 Q3 和 Q4 的问题。

#### D. 处理测距时间不匹配和消息丢失

根据前面的分析，Eq.(6)对于任意一对设备或机器人，都有一个测距周期，这个周期可能与其他对不同，因为距离  $d_2$  和速度  $v$  可能不同。换句话说，一个设备或机器人可能对它的每个邻居有不同的测距周期。例如，“远而慢”的邻居需要大的测距周期和低的测距频率，而“近而快”的邻居需要小的测距周期和高的测距频率。为了处理这些差异，一种简单的方法是选择最紧的测距频率要求(最小的测距周期)，并对所有满足所有要求的邻居进行设置。

$$P_A = \min_{Y \in \text{neighbors}} \left\{ \frac{e_0}{1 - e_0} \frac{d_{AY}}{v_{AY}} \right\}, \quad (8)$$

$d_{AY}$  和  $v_{AY}$  表示距离和速度。

因此，每个设备或机器人可能有自己的测距

周期(频率要求)由式(6)。因此，直接的问题(6)如下:如果测距对应有一个

不同的测距周期或如果消息丢失怎么办

传播(Q5)?这两个问题都涉及到不平衡的消息交换，我们在图 10 中总结为四种情况。在情况 1 中，A 接收到的比它发送的多。在情况 2 中，来自 A 的一条消息丢失。在情况 3 中，A 发送的比接收的多。在情况 4 中，来自 Y 的一条消息丢失。我们的群

测距协议必须能够处理所有情况。

在情况 1 和情况 2 中，消息  $Y_j$  将不携带 A 的任何接收时间戳，因为从  $Y_{j-1}$  发送以来没有收到任何消息。因此，在测距表中，接收更新后  $R_f$  将不存在，如表 II 中的灰色单元。对于这两种情况，我们的解决方案是丢弃  $Y_{j-1}$  和  $A_i$  的时间戳(如果存在的话)，如表 II 的下半部分所示，以便下一轮的范围可以继续。

情况 3 很容易处理。无论何时发送消息，我们总是更新/覆盖测距表的传输时间戳  $T_f$ 。

在情况 4 中，由于消息  $Y_j$  丢失，在  $A_{i+1}$  传输后，传输时间戳  $T_f$  被  $T_A$  覆盖。当 A 接收到消息  $Y_{j+1}$  时，存在不匹配。在 Y 端， $Y_j$  是最近传输的消息，所以时间戳是  $T_{Y_j}$  在消息  $Y_{j+1}$  中携带。这个时间戳

表二世

处理 1、2 例测距周期不匹配、消息丢失

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	$T_p = T_{A_{i-1}}$	$R_r = R_{Y_j}$	$T_f =$	$R_e =$

表三处理失配和损失情况

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

更新到测距表中，使  $T_r = T_{Y_j}$ 。但是在 A 端，由于  $Y_{j-1}$  是最后接收到的消息，所以测距表中保留了  $R_r = R_{Y_{j-1}}$ 。然后出现消息索引不匹配的情况，如表 3 中的灰色单元格所示。在这种情况下，我们的解决方案是丢弃  $A_{i-1}$ 、 $Y_{j-1}$  和  $Y_j$  的时间戳，如表 III 的下半部分所示，以便继续下一轮的量程。

我们通过过程 UpdateRX-M 总结这些步骤。

请注意，如果表 AY 不完整，计算过程将不会根据第 1 行中的 if 语句计算 ToF。至此，问题 5 得到了解答。

#### E. 处理密集邻居

根据过程生成，自上次传输以来的所有接收时间戳都包含在范围消息中。根据 Eq.(8) 将控制测距消息发送频率的测距周期设置为所有邻居的最小测距周期。因此，设备和

#### UpdateRX-M(table, msg)

```

1 UpdateRX-S(表、味精);
2 如果 msg 来自邻居 Y，那么
3   if  $R_f == \emptyset$  in table(AY) then
4      $R_r \leftarrow R_e, T_r \leftarrow \emptyset, T_f \leftarrow \emptyset, R_e \leftarrow \emptyset$  for table(AY);
5   else if  $\text{index}(T_r) \neq \text{index}(R_r)$  in table(AY) then
6      $R_p \leftarrow R_f, T_p \leftarrow T_f, R_r \leftarrow R_e$  for table(AY);
7      $T_r \leftarrow \emptyset, R_f \leftarrow \emptyset, T_f \leftarrow \emptyset, R_e \leftarrow \emptyset$  for table(AY);
8 结束

```

机器人交换测距消息的频率很高，每条测距消息中包含的接收时间戳的数量都很大，特别是当邻居密集时。

但是，消息大小受硬件或标准的限制。一旦有太多的邻居，一些时间戳最终必须被丢弃。回答问题 6 是至关重要的：如何选择时间戳以适应消息容量，以便可伸缩到密集的邻居？

为了在程序 generate 的选择中做出正确的决定，一个自然的想法是设置优先级，即，所需的测距频率越低，优先级越低。换句话说，“又远又慢”的邻居的时间戳会最先下降。然而，我们的实践表明，这种简单的优先级系统会导致饥饿问题，即高测距频率的邻居总是有机会交付它们的时间戳，而低频率的邻居由于它们的低优先级而几乎没有机会。

我们希望改进优先级系统的设计。我们通过一个综合的公交上车问题的例子来介绍我们的想法。一群乘客需要在汽车站乘坐同一辆公共汽车，所以他们共享公共汽车座位。每个乘客都会偶尔乘坐一次公交车，重复每次重复乘坐的时间。请注意，重复乘坐的时间可能因乘客而异。我们想安排有限的公共汽车座位

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

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



表四改进的测距表数据结构

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

乘客希望每个人轮流乘坐公交车，尽可能满足每个人的重复乘坐时间。

我们的基本想法很简单。我们为团队中的每一位乘客保持一个下一次想乘车的时间。一旦公共汽车到达车站，在车站等候的人就会根据他们的下一次乘车时间上车。下一个最急的想乘车时间的乘客是下一个上车坐下的人，以此类推。一旦所有的座位都坐满了，车就会开走，剩下的乘客会等下一辆车。每个上车的乘客更新他的下一个想要乘坐的时间到一个重复乘坐的时间后，从当前的时间。所有剩余乘客的下一次乘车时间不变。这样，当下一辆车到达时，上一辆车剩余的乘客和重复乘坐时间较少的乘客优先上车。而且，如果乘客很长时间没有出现在车站，我们就不再为他保留下次乘车时间。我们称这个方案为公共汽车上车方案。

设计了公交上车方案，改进了程序生成。读者可能已经猜到了，公共汽车代表测距消息，乘客代表邻居的接收时间戳。因此，乘客登上公共汽车表示邻居的接收时间戳被选择包括在接下来要广播的范围消息中。

我们现在改进了测距表的数据结构，如 table 所示。四。与表一相比，有三种新的表示法，即：， $P$ ,  $m$ ,  $ts$ ，其中  $P$  为邻居  $Y$  的最新测距周期，即，乘客  $Y$  的重复乘坐时间； $m$  表示邻居  $Y$  的下一个(预期的)交货时间，即，乘客  $Y$  下次想乘车时间； $ts$  为邻居  $Y$  的失效时间，即，乘客  $Y$  的停车维护时间。注意，对于每个邻居  $Y$ ,  $A$  维护一个范围表( $AY$ )。

就像在过程 Compute 中一样，在计算 ToF 之后更新测距周期。

步骤 6 计算(表( $AY$ ))

- 1 ToF = 计算(表( $AY$ ));
- 2 对表( $AY$ )用 Eq.(6)更新  $P$ ;
- 3 返回 ToF

一旦发送了包含邻居  $Y$  接收时间戳的范围消息，则更新下一个(预期的)发送时间  $t_n$ 。， $t_n$  应该比当前时间晚一个周期  $P$ 。而过期时间  $ts$  在收到邻居  $Y$  的消息后更新，即， $ts$  应该是比当前时间晚的过期时间。详细的伪代码可以在过程 UpdateTX 和 UpdateRX 中找到。

改进并适当维护了测距表之后，我们现在将重点放在如何生成测距消息上

步骤 7 UpdateTX(tables, msg)

- 1 UpdateTX-S(表、味精);
- 2 对于每个接收时间戳包含在 msg do 中的  $Y$
- 3  $t_n \leftarrow t_{current} + P$  表示表( $AY$ );
- 4 结束

UpdateRX(tables, msg)

- 1 UpdateRX-M(表、味精);
- 2 如果 msg 来自邻居  $Y$ ，那么
- 3  $ts \leftarrow t_{current} + T_{expiration}$  for table( $AY$ );
- 4 结束

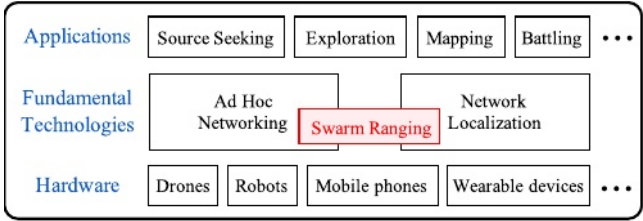


图 11 所示。群测距支持自组织网络和网络定位，这是任何群应用的基础。

根据各量程表中的下一个(预期)交货时间  $t_n$  和过期时间  $ts$ 。假设测距消息最多携带  $m$  个接收时间戳。选择原则很简单：对于还没有过期的邻居，我们按照下一个(预期的)交货时间排序，使它们按照时间顺序排列，并选择  $m$  个最紧急的邻居。为这  $m$  个邻居生成并发送一个带有接收时间戳的测距消息。然后，由过程 UpdateTX 更新  $m$  下一个交付时间。这个过程重复执行，在下一个(预期的)交付时间最近的时间生成下一个范围消息。Procedure Generate 显示详细信息。

步骤 9 生成()

- 1 为每个现有表( $AY$ )做
- 如果  $t_{current}$  为  $>$ ，则删除表( $AY$ );
- 3 结束
- 4 对所有表按  $t_n$  升序排序;
- 5 米:  $x \leftarrow \emptyset$ ;
- $m$  个表中  $Y$  的每一个邻居值是 6
- 7  $M_i^x \leftarrow M_i^x \cup (Y, R_Y)$
- 8 日结
- 束

在上述设计的帮助下，我们的协议现在处理密集邻居，这回答了 Q6。

F. 支持更高级别的协议和算法

对于任何成功的蜂群应用，例如，一组室内无人机搜索给定的目标，一群小型机器人探索并绘制未知的室内

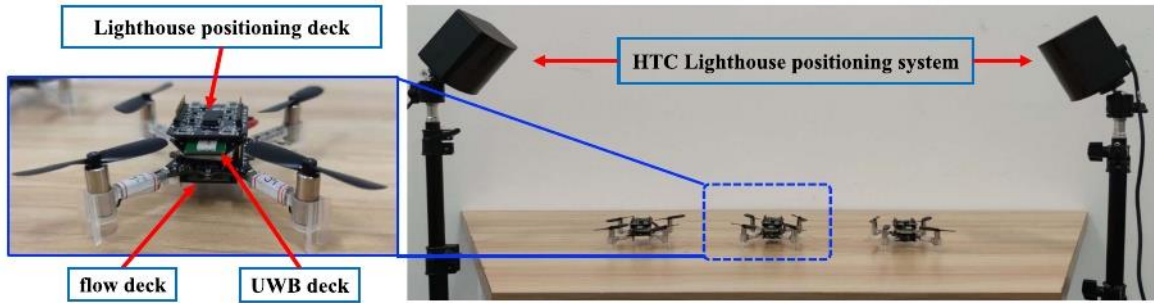


图 12 所示。实验中使用的硬件包括 crazyfly 无人机和 HTC Lighthouse 定位系统。

环境，一群有腿的机器狗在森林深处战斗，两项基本技术是 1)低延迟的自组织网络 和 2)实时网络定位。结构如图 11 所示。

群测距同时支持自组网和网络定位。(1)在自组织网络中，一些高级路由协议需要维护一个活动邻居列表。我们的群测距协议已经维护了这样一个列表(截止时间  $t_s$ )，可以直接使用。此外，有些路由协议会周期性地广播探测消息，例如 OLSR[16]中的 HELLO 消息，它可以与我们的测距消息相结合。(2)在网络定位中，使用节点网络来帮助定位成员，特别是节点之间的距离信息[15], [17]。我们的群测距协议为更高层次的定位算法提供了基本的距离信息。至此，第 7 个问题得到解答。

四、实现

我们在 STM32 微控制器驱动的微型无人机 Crazyflies 上实现了我们提出的蜂群测距协议，仅使用 192KB 内存和机载超宽带无线收发芯片 DW1000。在我们的实验中，无人机的数据是通过一台笔记本电脑通过工作在 2.4GHz 频段的通讯设备 Crazyradio PA 收集的。设有流动甲板，用于自动飞行控制和高度测量。

在一些涉及距离地面真值的实验中，使用了 HTC Vive Lighthouse 定位系统，该系统是一种光学室内定位系统，提供毫米级的位置数据作为地面真值。需要在无人机上安装一个额外的灯塔甲板来获取定位数据。图 12 是我们实验的硬件。左右两侧都有灯塔基站。桌子上放着三架“疯狂苍蝇”无人机，每架无人机都配备了灯塔甲板、顶部的超宽带甲板和底部的流动甲板。

所有无人机都使用同一个超宽带频道进行广播。我们将数据速率设置为 6.8 Mbps，并使用 128bit 的前置码。图 13 显示了我们实现的测距消息。在我们的实现中，范围消息由消息头和消息体组成。在消息头中，源地址字段用于标识范围消息的发送者。的消息

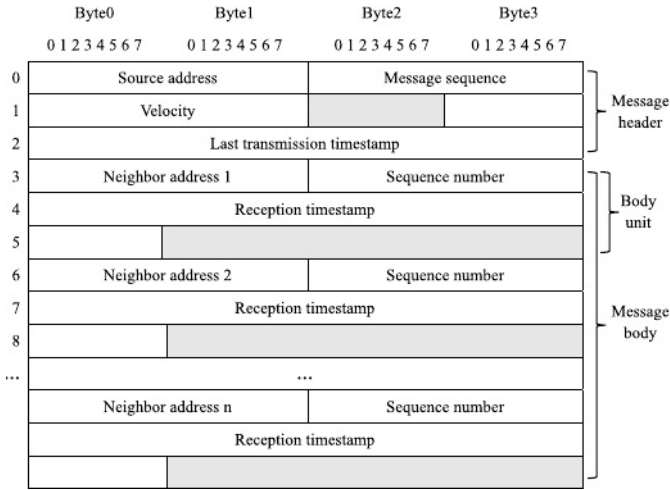


图 13 所示。我们实现的测距消息格式。

序列字段存储此消息的序列号。速度场和最后一次传输时间场是我们所提出的协议中所涉及的相应数据。消息正文由一组正文单元组成，每个正文单元对应一个邻居。一个主体单元携带来自邻居地址字段中声明的邻居的范围消息的标识和时间戳数据。最后一个传输时间字段与  $TX_{i-1}$  匹配。Mix 实现为消息体，速度场对应  $v$ 。

诉实验

本节通过实验对蜂群测距协议的各种性能进行评估。为了降低测距消息碰撞概率，我们将周期  $P$  随机分成两部分，即： $p + w$ ，底  $p$  是给定的固定实数  $p < p$ ，随机  $w$  是服从均匀分布的随机数  $w \in U(0, w)$ ，我们称  $w$  为随机窗口大小。所以我们 have  $\bar{P} = E(p + w) = p + W/2$ .

A. 测距周期和精度

为了评估测距周期对测距精度的影响，我们将两架无人机停在两个独立的椅子上，用

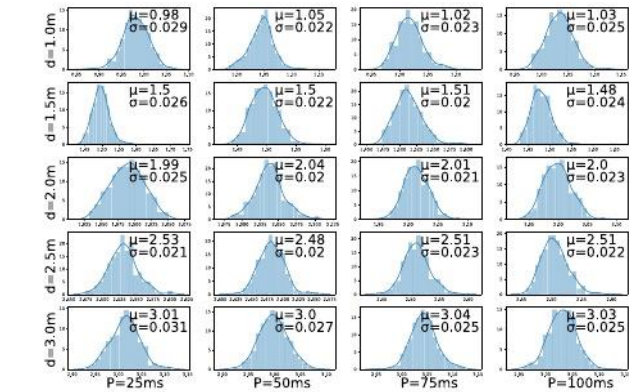


图 14 测距周期对测距精度的影响。  
表 v 消息接收比和测距比

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%

视线清晰，且距离任何墙壁 1 米以上。我们禁用了量程调整，并设置  $W = 0$ 。

研究了步长为 25ms 的不同测距周期  $P \sim 100\text{ms}$ ，步长为 0.5m 的 1.0 ~ 3.0m 不同距离  $d$  下的测距精度。在每个设置下，测距结果遵循一定的随机分布。其概率密度函数如图 14 所示。可以看出，在不同距离下，测距周期  $P$  不影响测距精度。

B. 消息接收比和测距比

本小节利用 4 架无人机组成一个小蜂群来评估信息接收比和测距比。A、B、C、D 4 架无人机静置近距离，保证良好的无线信道。我们为所有无人机设置了平均射程周期  $P = 50\text{ms}$  ( $P = 30\text{ms}$ ,  $W = 40\text{ms}$ )。每架无人机总共传送 6000 条测距信息。

我们定义消息接收率为接收到的消息数与发送的消息数之比，测距率为测距计算成功的次数与发送的消息数之比。表 V 为 a 无人机记录的次数和比率，可以看出，平均消息接收率超过 90%，平均测距率超过 70%。测距比低的原因是测距周期是随机的，这导致了周期随机错配，即：，图 10 中的 Case 1 和 Case 3，从而减小了测距比。

C. 速度和测距精度

本节评估速度  $v$  对测距精度的影响。HTC Lighthouse 室内定位系统用于提供距离地面真实度。

由式(5)可知，测距周期  $P$  和无人机速度  $v$  都会影响测距精度。因此，我们分别研究它们。

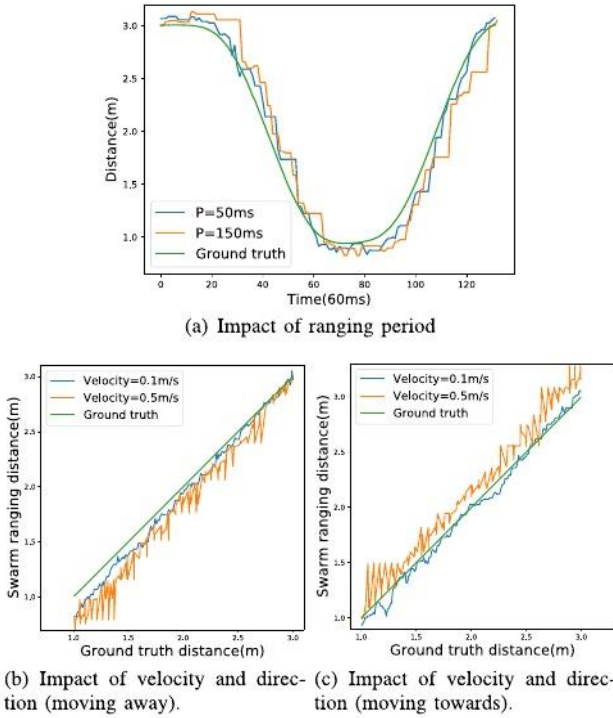


图 15 所示。不同速度和测距周期的测距精度。

首先，利用 A、b 两架无人机评估测距周期的影响。A 无人机在固定位置盘旋。无人机 B 首先飞向它，从 3 米外的位置开始，两架无人机高度相同，均为 100 厘米。B 在距离 A 1 米的地方停下来，然后飞回起始位置。B 的飞行速度设为  $v = 0.5\text{m/s}$ 。重复 2 次，测距周期分别为  $P = 50\text{ms}$  和  $150\text{ms}$ 。测距结果如图 15(a)所示。很明显，当  $P$  变大时，准确率下降。我们还观察到测距结果相对于地面真实值有延迟，且  $P$  越大，延迟越大，这与我们在 Eq.(5)中的理论分析是一致的。

其次，在相同的设置下考察了飞行速度的影响。不同之处在于，我们现在将射程固定为  $P = 50\text{ms}$  ( $P = 45\text{ms}$ ,  $W = 10\text{ms}$ )，但为无人机 B 设置了两种飞行速度，0.1m/s 和 0.5m/s。我们将两个飞行方向分开，分别朝向和远离，如图 15(b)(c)所示。飞行速度越快，采集的采样测距数据越少，因为飞行时间越短。为了获得足够的样本点，设置无人机 B 以 0.5m/s 的速度飞行 5 次。从图 15(b)(c)可以看出， $v = 0.1\text{m/s}$  时的测距结果比 0.5m/s 时更稳定。此外，(b)(c)也揭示了(a)的相同结论，测距有延迟。

D. 不匹配测距周期的性能

本实验共有 4 架无人机，分别为 A、B、C、D，我们设置 A、B、C、D 的平均测距周期  $P = 30\text{ms}$ 、 $40\text{ms}$ 、 $50\text{ms}$ 、 $60\text{ms}$ ，其中所有无人机的随机窗口大小设为  $W = 40\text{ms}$ 。图 16 为无人机 A 在 200s 持续时间内记录的测距计数。可以看出，我们的蜂群测距协议平滑地处理了测距周期不匹配的问题，即较短的距离



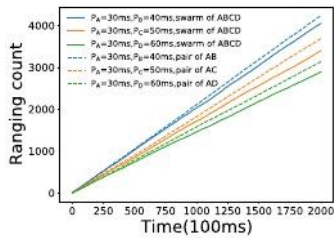


图 16 所示。不匹配测距周期的性能。

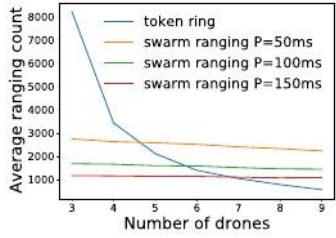


图 17。与基于令牌环的测距比较。

周期越长，发生的幅度越大。在不受干扰的情况下，将该对移出蜂群，距离计数增加。

E. 与基于令牌环的测距比较

我们将我们的群测距协议与基于令牌环的测距算法进行了比较。我们改变平均周期  $P$ ，从 50ms 到 150ms，同时保持随机窗口大小  $W = 80ms$ 。我们改变无人机的数量，从 3 架到 9 架。图 17 为无人机 A 记录 200 秒的测距计数。

图 17 中的任意一点为无人机 A 与相邻无人机的平均测距成功次数。我们可以看到令牌环算法的性能随着参与者的增长急剧下降，因为它是顺序执行的。而群测距的性能略有下降，因为随着数量的增加，消息冲突的概率会增加。当无人机数量超过 5 架时，我们的群测距协议性能优于令牌环算法。当 9 架无人机参与时，根据我们的协议，一架无人机成功与另一架无人机进行航程的平均数量约为令牌环算法的 5 倍。

F. 测距周期的适应性

在本节中，我们参照公式(6)来评估速度  $v$  和距离  $d$  都可以影响周期  $p$  的适应性，我们不设置固定的  $p$  和  $W$ ，而是根据公式(6)来调整测距周期  $p$ 。HTC 灯塔系统用于定位和距离测量。

我们进行了两个实验。在第一个实验中，无人机 A 在固定的位置盘旋，无人机 B 以固定的速度  $v$  围绕无人机 A 旋转，图 18(A)为无人机 B 在 100 秒的持续时间内记录的测距计数。我们以步长 0.5m 改变距离  $d$  到 1.5m，以步长 0.1m/s 改变无人机 B 的速度  $v$  到 0.5m/s。如图 18(a)所示，距离计数随着速度的增加而增加。此外，随着距离的增加，测距计数下降。在第二个实验中，无人机 A 和

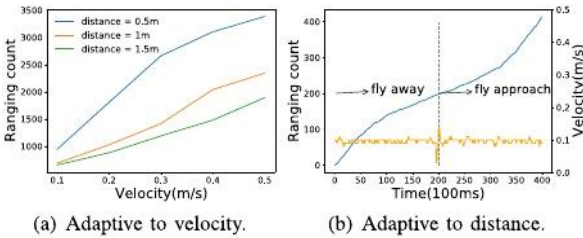


图 18 所示。测距周期与速度和距离相适应。

B 悬停在固定位置，彼此靠近。然后无人机 B 以 0.1m/s 的速度远离无人机 A，飞行 20 秒后返回。图 18(b)是无人机 A 记录的测距计数和无人机 b 的速度。可以看出，随着无人机 b 的飞走，测距计数的增长速度变慢，随着无人机 b 的靠近，测距计数的增长速度加快。两个实验的结果都验证了该协议对测距周期的适应性。

G. 密集群公交上车方案的性能

IEEE 标准 802.15.4z-2020 将报文大小限制为 127 字节[13]。作为包负载，范围消息的大小限制为 100 字节，因为必须减去包头。因此，主体单元的数量(每个相邻单元一个)被限制为 7 个，如图 13 所示。

在这个实验中，我们评估了在群中邻居密集的情况下，为群测距协议提出的公交上车方案的性能。我们将 11 架无人机彼此靠近，并限制每个测距信息最多携带 7 个身体单位，因此在测距信息中不可能包含所有邻居收到的所有时间戳。为了评估该方案在下一个测距消息中选择邻居的性能，我们将其与两种简单的基线方法进行了比较。第一种基线方法不考虑优先级，即:在美国，当必须进行选择时，邻居的选择完全基于他们的身份(ID)号码。第二个基线方法提出了一个简单的优先系统，即:，所需的测距频率越高，优先级越高，邻居按此顺序选择。我们进行了两次实验，分别对这两个基线进行比较。

在第一个实验中，我们设置每架无人机的平均测距周期  $P^- = 50ms$  ( $W = 0ms$ )，并记录 200 秒内的平均测距次数。结果如图 19 所示。我们可以看到，通过第一种基线方法，ID 大于 8 的无人机的测距计数要少得多。这是因为，这些无人机执行测距的机会非常少，因为测距信息只能携带 7 个邻居的测距信息(体单元)。相反，我们的上车计划保证所有邻居都有公平的机会去游览。从图中我们可以看到，所有无人机的测距次数都没有太大的差异。

在第二个实验中，我们对不同的邻居设置不同的测距周期。对于无人机，我们设置  $P^- = 50ms$ ，对于无人机 2, 65 ms，对于无人机 3 设置 80ms，以此类推。第二基线采用简单优先系统，使无人机的测距周期更大(所需的测距频率更小)。

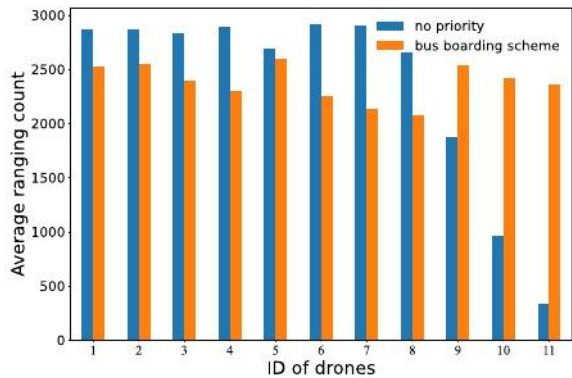


图 19 所示。资 公交上车方案平衡了行程分配

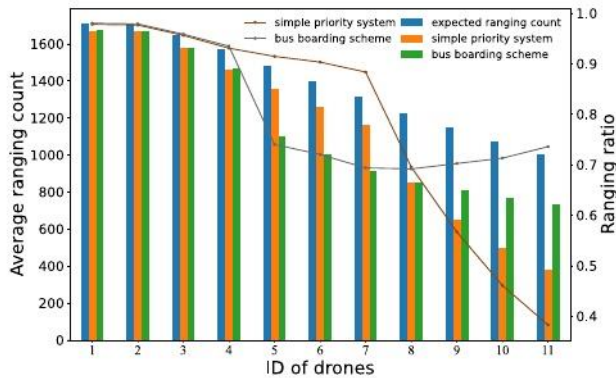


图 20。公交上车计划保证了每架无人机有公平的机会进行定位。

有较低的优先级来竞争范围消息。结果如图 20 所示，其中期望距离计数为距离计数的理论上界，即，没有消息丢失和一个非常大的包大小，允许所有邻居范围。

测距比(右 y 轴)是测量值与期望值的比值。从图中我们可以看出，在简单优先级系统中，当 ID 号大于 7 时，低测距频率无人机的测距比急剧下降。作为比较，在我们的解决方案中，所有无人机的测距比都超过 69%，这表明我们的公交上车方案确保了所有邻居之间公平竞争测距资源的机会。

h. 演示实验

为了测试实时测距精度，我们进行了避碰实验。可以在 <https://www.youtube.com/watch?v=jT3mPQvHadM> 上找到演示视频。

在这个实验中，8 架 crazyfly 无人机在一个小于 3m × 3m 的紧凑区域内，在 70cm 的高度悬停。而第九架疯狂苍蝇无人机则是由人工控制飞进这个区域的。我们将平均测距周期  $P = 100ms$  ( $P = 60ms$ ,  $W = 80ms$ )和最大测距消息体单位(每个邻居一个)设置为 8。在演示视频和图 21 中，通过群体测距协议，无人机通过测距距离检测到飞来的无人机，当距离小于阈值 30cm 时，无人机会降低高度以避免碰撞。



图 21。 疯狂飞行的无人机在紧凑飞行时避免碰撞。

六. 相关工作

超宽带被广泛应用于[18]-[24]的室内定位。Tiemann 等人[18]和 Corbalan 等人[19]设计了基于 TDOA 的测距算法，需要时间同步。Xu 等人[20]融合视觉、惯性和超宽带信息进行空中群定位。同时，Cao 等人[21]只使用单一的 UWB 锚(基站)对机器人进行定位。此外，Poulouse et al.[22]使用长短期记忆(LSTM)网络通过超宽带(UWB)来预测用户在室内场景中的位置。Yu 等人[23]提出了一种不依赖环境、不依赖先验知识的测距 NLOS 识别和缓解方法，并制定了一种规则来选择合适的 NLOS 范围进行位置估计。Macoir 等人[24]设计了一种基于超宽带的定位系统，使用基础设施锚节点。Guizar 等人[25]研究了运动捕获系统中 MAC 调度策略对超宽带测距精度的影响。Courtay 等人[26]提出了一种在无线体域网络中实现室内定位和运动跟踪的平台。除了超宽带，还有其他技术被用于群网络的定位。例如，Rivard 等人[27]开发了一种基于超声脉冲飞行时间估计的相对定位系统。De Silva 等人[28]提出了一种基于超声的三维传感器节点用于多机器人定位。Wu 等[29]提出了一种基于 ZigBee 锚节点的机器人网络设备精确定位方法。同时，McGuire 等[30]将蓝牙应用于微型飞行器群的相对定位和避碰;Khanh 等人[31]提出了一种基于 Wi-Fi 的室内定位和导航方法，该方法使用了基于云计算的云计算系统。

近年来，在超宽带研究领域[32]-[35]中，并行测距正在成为一种新兴的趋势。Corbalan 等人[32]，[33]提出利用信道脉冲响应判别同一请求的重叠应答的个体到达时间，称为并发范围。Heydariaan 等人[35]后来将这一想法扩展到反射弹性。而 Stocker 等人[34]对其进行了扩展，以支持无限数量的标记。

也有一些研究聚焦于大数字[19]，[34]，[36]或高机动性[37]，[38]的超宽带群。Corbalan 等人[19]和 Stocker 等人[34]通过 TDOA 或并发测距来定位无数的标签。他们的工作致力于锚-标签模型，不适用于群体场景。Cao 等人[36]提出了一种新颖的时分多址(TDMA)算法，该算法可以在由 UWB 标记节点组成的大型网络中快速调度 UWB 介质的使用。Risset 等人[37]研究了快速运动的超宽带群测距问题

只有两个超宽带标签。Liu 等[38]提出了一种用于自主室内移动机器人的 INS/UWB 组合定位的有效系统框架。

超宽带也被用于[39]、[40]的数据通信。Mohammadmoradi 等人[39]提出在超宽带网络中同时测距和通信。Vecchia 等人研究了超宽带的并发传输问题。

新修订的 IEEE 标准 802.15.4z-2020[13]提出了基于 SS-TWR 或 DS-TWR 的多节点测距解决方案。这些解决方案总是涉及多个范围阶段和角色(即., 控制器, 被控器, 启动器, 响应器)来定义一个测距程序。一个测距过程由多个测距回合组成, 而一个测距回合又由多个阶段组成。每个阶段都是一个足够持续的时期, 让某个角色达到某个目的。例如, 测距控制阶段通常是一个测距轮的第一阶段, 以便控制器发送决定测距参数的控制消息。虽然应用广播, 计划阶段需要一个集中的测距过程。这些解决方案不仅复杂, 而且难以配置。

综上所述, 目前的相关工作还没有涉及到针对动态密集网络的测距协议的设计。

7 结论与未来工作

本文提出了一种针对动态密集机器人或设备网络的超宽带群测距协议。其基本思想是设计周期性广播的测距信息。时间戳由该消息携带, 以便计算距离。我们的群测距协议简单而高效, 因为只有一种类型的消息;由于测距周期适应了测距对的速度和距离, 并对消息丢失进行了适当的处理, 因此具有自适应和鲁棒性;它是可扩展和兼容的, 因为一个公共汽车上车方案被设计来处理密集的邻居, 并支持更高级别的网络和本地化协议和算法。最后, 以 STM32 微控制器驱动的微型无人机 Crazyflies 实现了该协议, 内存仅为 192KB, 机载 UWB 无线收发芯片为 DW1000。得益于蜂群测距协议, 9 架“疯狂苍蝇”无人机在狭小空间飞行时可以自动避免碰撞。

在未来, 我们还可以对现有的群测距协议的设计和实现进行进一步的改进。首先, 可以采用数据挖掘技术, 利用网络产生的大量测距数据来实现更高的测距精度。第二, 以更通用的方式实现该协议, 与智能手机、可穿戴设备等进行交互, 形成更大的网络, 并设计数据融合算法, 结合多种类型的传感器数据, 减少测距误差。第三, 在拥有更快 CPU 和更大内存的高级硬件上实现该协议, 以支持拥有数百个设备或机器人的更大网络。第四, 所提出的测距资源分配算法可以根据应用任务的要求进行调整或改进。最后, 一些可能的增强

在动态密集的网络环境下, 研究丢包率与测距频率之间的权衡问题。

鸣谢

作者感谢密苏里大学堪萨斯城分校(UMKC)沈晓军教授、匿名审稿人和编辑们的建设性意见和宝贵建议, 这对提高论文的质量和表达能力起到了很大的帮助。

参考文献

[1] crazyfly 2.1 - bit 狂热商店。访问日期:2021 年 7 月 22 日。(在线)。可用:<https://www.bitcraze.io/2019/02/the-crazyfly-2-1-is-here/>

[2] RoboMaster EP-DJI。访问日期:2021 年 7 月 22 日。(在线)。可用:<https://www.dji.com/robomaster-ep>

波士顿动力公司推出 SPOT 机器人商业销售。访问日期:2021 年 7 月 22 日。(在线)。可用:<https://www.bostondynamics.com/press-release-spot-commercial-launch>

[4] Z. Zhakypov, K. Mori, K. Hosoda and J. Paik, “设计最小和可扩展的昆虫启发的多运动微型机器人”, 《自然》, 第 571 期。7765, 第 381-386 页, 2019 年 7 月。

余杰, 金丹, 金凯福。陈强, 王强, 袁凯, 张磊, “生物流体中微机器人群的主动生成和磁驱动”, 《自然通讯》。第 10 卷, no. 1, 2019 年 12 月 1 - 12 页。

杨洋, M. A. Bevan, “胶体群捕获和运输货物”, 科学与技术。副刊, 第 6 卷, no. 2020 年 1 月 4 日, 艺术。不。eaay7679。

[7] K. N. McGuire, C. De Wagter, K. Tuyls, H. J. Kappen 和 G. C. H. E. De Croon, “一群微型飞行机器人探索未知环境的最小导航解决方案”, Sci. 机器人。第 4 卷, 没有。2019 年 10 月 35 日, 艺术。不。eaaw9710。

曾杰, 李振中, 罗杰, 单飞, 吴伟, “超宽带群测距”, 计算机科学与技术学报。Commun., 2021 年 5 月第 1-10 页。

[9] G. Coulouris, J. Dollimore, T. Kindberg 和 G. Blair, 分布式系统:概念和设计。伦敦,英国:皮尔森,2012 年。

[10] DW1000 用户手册, Decawave Ltd, 都柏林, 爱尔兰, 2017。

[11] SmartTag Plus-Samsung。访问日期:2021 年 6 月 25 日。(在线)。可用 :<https://www.samsung.com/us/mobile/mobileaccessories/phones/galaxy-a-smart-tag-1-pack-black-ei-t7300bbegus/>

[12] AirTag-Apple。访问日期:2021 年 6 月 25 日。(在线)。可用:<https://www.apple.com/airtag/>

[13] IEEE 低速率无线网络标准-修订 1:增强超宽带(UWB)物理层(PHYs)和相关测距技术, IEEE 标准 802.15.4z-2020, 2020, pp. 1 - 174。

[14] B. Broecker, K. Tuyls 和 J. Butterworth, “袖珍无人机上基于距离的多机器人协调”, 在 Proc. IEEE Int. 相依的机器人。奥特曼。(ICRA), 2018 年 5 月, 第 6389-6394 页。

[15] S. Li, M. Coppola, C. De Wagter, 和 G. C. H. E. De Croon, “基于距离的相对定位的微型飞行机器人自主群”, 2020,arXiv:2003.05853。

[16](2003)。优化的 OLSR (Link State Routing Protocol)。互联网请求评论, RFC 编辑器, RFC 3626。(在线)。可用:<https://tools.ietf.org/html/rfc3626>

戴伟, 沈勇, 温明志, “网络定位与导航策略研究”, 《计算机应用与研究》, 第 1 期。7, 1224-1254 页, 2018 年 7 月。

[18] J. Tiemann, Y. Elmasry, L. Koring 和 C. Wietfeld, “ATLAS FaST:用于可靠的超宽带定位的快速和简单的预定 TDOA”, 在 Proc. Int. 相依的机器人。奥特曼。(ICRA), 2019 年 5 月, 第 2554-2560 页。

[19] P. Corbalón, G. P. Picco 和 S. Palipana, “合唱:超宽带并发传输用于类似 gps 的无源定位无数目标”, 在 Proc. 18 Int. 相依 Inf. 过程。传感器 Netw., 2019 年 4 月, 第 133-144 页。

徐洪华, 王磊, 张勇, 邱凯, 沈 s., “基于分散视觉-惯性-超宽带融合的空蜂群相对状态估计”, 中国信息工程学报。相依的机器人。奥特曼。(ICRA), 2020 年 5 月, 8776-8782 页。

[21] Y. Cao, C. Yang, R. Li, a. Knoll, G. Beltrame, “基于单一超宽带锚点的精确位置跟踪”, 《电子与电子工程学报》。相依的机器人。奥特曼。(ICRA), 2020 年 5 月, 第 2344-2350 页。

A. Poulou 和 D. S. Han, “使用深度学习 LSTM 网络的超宽带室内定位”。科学。第 10 卷, no. 18, 6290 页, 2020 年 9 月。



李勇, 张胜, 张凯, 张凯, 张凯, “室内环境下超宽带定位的一种新型非  
los 抑制算法,” *阿明费. 抛光工艺*. 第 68 卷, no. 1, 第 686-699  
页, 2019 年 1 月。

[24] N. Macoir 等人, “基于电池供电的无线骨干网的无人机库存管理超  
宽带定位”, *《传感器》*, 第 19 卷, 第 1 期. 3, 467 页, 2019 年 1  
月。

A. Guizar, A. Ouni, C. Goursaud, N. Amiot 和 J.-M. Gorce, “在超宽带体域网  
络中, MAC 调度对运动捕捉定位精度的影响”, *《第 9 国际程序》*. 主  
体区域网络., 2014, 第 1-8 页。

[26] A. Courtay, M. Le Gentil, O. Berder, A. Carer, P. Scalart 和 O. Sentieys,  
“Zyggie:用于室内定位和运动跟踪的无线身体区域网络平台”, 在  
Proc. IEEE Int. 计算机协会. *电路系统. (ISCAS)*, 2018 年 5 月, 第  
1-5 页。

[27] F. Rivard, J. Bisson, F. Michaud, D. Letourneau, “多机器人系统的超声相  
对定位”, 在 Proc. IEEE Int. 相依的机器人. *奥特曼*., 2008 年 5 月,  
第 323-328 页。

王志强, 王志强, 王志强, “一种基于超声和视觉的多机器人相对定位传  
感器”, *《电子与信息学报》*, vol. 15, no. 7. 3, 1716-1726 页, 2015  
年 3 月。

吴旭东, 吴宏, 辛斌, 陈晨, “基于多维尺度和无线传感器网络的群体机器  
人精确协调定位”, *《计算机科学与工程学报》*. 相依 Netw., *Sens.  
Control (ICNSC)*, 第 1 卷, 2021 年 12 月, 第 1-6 页。

[30] K. McGuire, M. Coppola, C. de Wagter 和 G. de Croon, “面向现实环境  
中多口袋式无人机的自主导航”, 在 Proc. IEEE/RSJ Int. 相依智能。  
*机器人系统. (IROS)*, 2017 年 9 月, pp. 244-249。

程德光, 阮文华, 陈志强. 范教授.E.-N. 嗯, “Wi-Fi 室内定位和导航:  
基于云的云计算方法”, 哼。为中心第一版。正. *科学*. 第 10 卷,  
no. 1, 2020 年 12 月 1-26 页。

[32] P. Corbalán 和 G. P. Picco, “超宽带无线电的并行测距:实验证据、挑战  
和机遇”, *《Proc. EWSN》*, 2018 年, 第 55-66 页。

[33] P. Corbalán 和 G. P. Picco, “超宽带并发测距”, *ACM Trans. 传感器  
Netw*. 第 16 卷, 没有. 4, 2020 年 11 月 1-41 页。

[34] M. Stocker, B. Großwindhager, C. A. Boano, 和 K. Römer, “SnapLoc:一个  
基于超快 u wlan 的室内定位系统, 无限数量的标签:演示摘要,” 在  
Proc. 18 Int. 相依 Inf. 过程. *传感器 Netw*., 2019 年 4 月, 第 61-72  
页。

[35] M. Heydariaan, H. Mohammadmoradi 和 O. Gnawali, “R3:用超宽带无线  
电反射弹性并行测距”, 第 15 国际程序. 相依 Distrib. 第一版. *传感  
器系统. (dcos)*, 2019 年 5 月, 第 1-8 页。

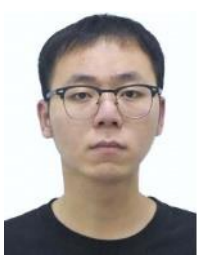
曹勇, 陈晨, D. St-Onge, G. Beltrame, “基于分布式 TDMA 的移动超宽带  
网络定位”, *《电子与信息学报》*, vol. 8, no. 1. 17, 13449-13464 页,  
2021 年 9 月。

T. Risset, C. Goursaud, X. Brun, K. Marquet 和 F. Meyer, “快速运动的超宽  
带测距”, 在 Proc. Int. 室内定位室内导航. *(IPIN)*, 2018 年 9 月,  
第 1-8 页。

刘建峰, 濮建新, 孙立峰, 何振山, “一种基于 INS/UWB 的自主室内移动  
机器人鲁棒集成定位方法”, *传感技术*, vol. 19, no. 7. 4, 第 950  
页, 2019 年 2 月。

[39] H. Mohammadmoradi, M. Heydariaan 和 O. Gnawali, “SRAC: UWB 网络  
中的同步测距和通信”, 第 15 国际程序. 相依 Distrib. 第一版. *传感  
器系统. (dcos)*, 2019 年 5 月, 第 9-16 页。

D. Vecchia, P. Corbalan, T. Istomin 和 G. P. Picco, “玩火:探索超宽带无线电的  
并发传输”, 在 Proc. 16 Annu. *IEEE Int. Commun 相依 Sens., Netw.  
(SECON)*, 2019 年 6 月, 第 1-9 页。



霍浩东(研究生成员, IEEE), 于 2021 年获得河南大学  
计算机学士学位, 中国开封。他目前在东南大学  
攻读网络科学与安全硕士学位。主要研究方向为  
群体智能、无人机网络、算法设计。



曾嘉欣, 2016 年获合肥工业大学计算机学士学位,  
2021 年获东南大学计算机技术硕士学位。他目前是  
中国银联有限公司的软件工程师。



李增宝 2019 年获安徽工业大学理学学士学位, 2022  
年获东南大学理学硕士学位。主要研究方向为无线  
网络协议和算法设计。



吴伟伟(IEEE 会员), 2006 年获华南理工大学计算机  
科学学士学位, 2011 年获香港城市大学与中国科学技术  
大学联合获得计算机科学博士学位。2012 年赴新加  
坡南洋理工大学数学系进行博士后研究。现任东  
南大学计算机科学与工程学院教授。

他的研究兴趣包括优化和算法分析、众包、强化学习、博弈论、无线通信  
和网络经济学。



峰山(IEEE 成员)2015 年获东南大学计算机科学博士学  
位, 南京, 中国。2010 年至 2012 年, 他在美国密  
苏里州堪萨斯城大学计算机与工程学院访问学习。  
现任东南大学计算机科学与工程学院副教授。主  
要研究方向为物联网、无线网络、群体智能、算  
法设计与分析等。



罗俊洲(IEEE 会员)分别于 1982 年、1992 年和 2000 年  
获得东南大学应用数学学士学位和计算机网络硕士学  
位和博士学位。东南大学计算机科学与工程学院正  
教授。主要研究方向为下一代网络架构、网络安  
全、云计算和无线局域网。他是 IEEE 计算机学会  
的成员和 IEEE SMC 计算机支持技术委员会的联  
合主席

合作设计。他是 ACM 的成员和 ACM 中国 SIGCOMM 主席。