

A new approach to realize drone swarm using ad-hoc network

Omar Shrit*, Steven Martin*, Khaldoun Al Agha*, Guy Pujolle†

* LRI, Laboratoire de Recherche en Informatique, Univ. Paris-Sud/CNRS, Paris-Saclay University, France

shrit, smartin, alagha@lri.fr

†LIP6, Laboratoire d'Informatique de Paris 6, UPMC/CNRS, Sorbonne University, France

guy.pujolle@lip6.fr

Abstract—Swarms of drones are increasingly being requested to carry out missions that cannot be completed by single drones. Particularly in the field of civil security, strong needs emerge in terms of surveillance and observation of hostile, distant or extended areas. Currently, existing solutions do not meet this demand, as they are generally based on too heavy infrastructures or over-processing, without consideration of quality of service (especially in terms of throughput). In this paper, we propose a light and efficient solution to synchronize and orchestrate a swarm of drones, based only on ad hoc communications to position drones. Our proposal operates a swarm by human piloting a drone (the leader), while all the others (the followers) are completely autonomous and follow the leader using Wi-Fi signal strength. We provide algorithms to realize a basic line scenario. Based on two software developed in the framework of this work, real flight tests have been conducted and experiment results are shown.

Index Terms—drone swarm, ad hoc network, leader-follower formation

I. INTRODUCTION

Due to the technological advancement of the last years in hardware and software in the field of robotics, the use of the term “drones” has moved from military sector to hackers and developers. Nowadays, civil unmanned aerial vehicles have a promising future in many sectors. By making these robots intuitive, friendly, easy to use and even manufactured by a basic 3D printer, hundreds or thousands of potential applications and services are made possible. With this rapid growth, most markets have been flooded by all kinds of drones, from fixed-wing aircrafts to multirotor crafts or a combination of the two. Beyond the known set of applications such as monitoring and delivery, these vehicles can perform very difficult tasks due to their high maneuverability (e.g. acrobatic capacities [1]) and their high level of stability, even losing a propeller during their flight time [2].

However, there are situations where a single drone is not enough, because its capacities (e.g. radio range, zone coverage, sensors) are not suitable for the application or the mission requested. Some works wish to overcome these limitations by using a swarm of drones. For example, a group of drones could collaborate to build tensile structures [3] or even buildings. It could also be used for shows and choreographies [4]. Currently, strong needs emerge in the field of civil security

for observation in hostile, difficult or large areas, as well as for monitoring of sites (e.g. borders, seas, forests) and populations during specific events. But existing solutions are inadequate. Indeed, in order to realize a swarm of drones, another problem arises, which is the need for a coordination system that provides the exact location of a drone in space, whether in an indoor or an outdoor environment.

Because of these limitations, most current research uses relative location techniques, which require the deployment of pre-flight infrastructure (motion capture system, Indoor Positioning System, etc.). In this paper, we show that it is possible to operate a swarm of drones using a single remote control without any other equipment than the drones themselves. Without relying on any existing infrastructure or any specific sensors such as the GPS module that is imprecise and inefficient in certain situations (e.g. indoor), we propose a solution to synchronize and orchestrate a set of drones based on ad hoc communications.

Indeed, by forming an ad hoc network (based on IEEE 802.11 in our experimentation), we can determine the distances between drones according to adequate data rates that depend on received signal strengths. So, no position and no distance is predetermined. Drones position themselves compared to their neighbors based on the received signal powers from them, indicating perfectly the quality of the application. To clarify the idea, we use the analogy of two individuals bringing closer, or moving away from each other to discuss, depending on the noise in the environment. Drones are consistently adapting their relative positions to maintain the same received signal power, thus ensuring the requested application level in terms of data rate.

Our solution is based on this concept and implements a new effective mechanism to maintain a swarm of drones using only ad hoc communications, while continuously ensuring the quality of the service needed to complete the mission.

The following paper is organized as follows. Related work on drone swarms is presented in Section II. Our solution in terms of architecture and algorithm is described in Section ???. In Section IV, we talk about our software implementation and the proposed routing technique. In Section V, the results of a real test flight are shown to illustrate the relevance of our solution. Finally, we conclude in Section VI and discuss some aspects for future work.

II. RELATED WORK

In this section we summarize the main works related to swarms of drones, dividing the state of the art into two parts that concern the fundamental question: How to position drones relative to others.

A. With a localization system

To clarify the notion of existing localization systems, the following two platforms have been developed:

- **Flying Machine Arena** [5] It is a distributed system, developed at the University of ETH, characterized by a robust and reliable infrastructure at hardware and software levels. This platform consists of a motion capture system, where Vicon cameras measure the vehicle's position and altitude at a frequency of 200 Hz. Then, this data is sent using multicast UDP to software modules. These are distributed among several standard computers interconnected using Ethernet or a dedicated wireless channel. Furthermore, the platform provides several features such as recording, playback, vehicle independence, simulation and hardware abstraction. This platform has shown an impressive performance and has been used in a large number of demonstrations such as ball juggling [6], pole acrobatics [1] and cooperative ball throwing-catching [7].
- **GRASP** [8] As in flying machine arena, this platform depends on Vicon capture system to localize the drones. Nevertheless, software architecture is totally different since it depends on Finite-State Machine (FSM) and uses Robot Operating System (ROS) to integrate the control software into the motion system. The use of FSM is intended to facilitate the conduct of experiments and handle failure. Indeed, it succeeds in executing the appropriate sequence of behaviors requested by the user. Using this platform, authors of [9] also managed to realize a swarm of micro-quadcopters. A completely centralized approach has been used to reduce the complexity of coordinating a large number of quadcopters. They divided the drones into a given set of groups, each one containing a number of labeled drones. Groups collaborate with others to realize specific tasks.

Apart from these indoor testbeds, there have been other proposals to realize swarms of drones. In [10], the authors propose to use an ultra-wideband (UWB) localization system where several anchors (radio modules placed at known locations) periodically send UWB signals to drones. Independently, each drone uses an embedded TOA (Time Of Arrival) or TDOA (Time Difference Of Arrival) technique to locate itself in relation to the anchors. Due to the use of one-way communications from the anchors to any drone, this method might be ideal for a drone fleet. Similarly, in [11] they study the effect of a ground station equipped with a large number of antennas (MIMO)

which communicates with a swarm, each drone having a single antenna. The study deals with the relationship between the number of antennas used and the uplink capacity with all drones.

Drones could also use Simultaneous Localization and Mapping (SLAM) techniques to discover their surrounding environment [12]. This requires a set of low-cost on-board sensors such as monocular cameras, IMU, or RGB-D to map the nearby. As a result, each drone obtains a map that can be shared with others or fused using a ground center. Once the global map is constructed, it could be used to localize each drone in the environment and thus provide a relative localization system.

However, none of the above is suitable for our purpose, since they require the installation of heavy infrastructure.

B. Without localization system

Thanks to developments in optical flow technology, an on-board optical sensor (e.g. Px4 sensor [13]) can be used to stabilize the drone in the air. This technique substitutes the need for an external localization system. In general, most state of the art algorithms that rely on optical flow use visual relative localization to realize drone swarm. In [14], the authors simulate a tractor-trailer principle by using two drones: a leader L and a follower F . There is no communication between drones, each one being equipped only with a limited range camera. By assuming that visibility is always maintained between them, F uses its camera to track the target L . Simulations show the value of their motion algorithm. In [15], the authors verify this algorithm using real-time flight in indoor and outdoor environment, without using any motion or GPS system. This kind of system replaces the need for a localization system by using visual pattern recognition.

Instead of using heavy infrastructure, these solutions are based on complex sensors, require centralized algorithms and overprocessing.

We present in the next section a light, distributed and efficient solution to manage a swarm of drones, only based on ad hoc network.

III. PROPOSED SOLUTION

A. System Architecture

Among the huge set of applications for drones, there is a real interest in video surveillance and observation. A fixed wing aircraft drone or a quadcopter can be operated to realize such a mission. But the possibilities are limited, in the sense that the covered area is reduced (because of radio range and the characteristics of a single drone). To cover an extended area or allow long-distance real-time video, we propose a basic drone swarm where we use the notion of leader and followers. The drones and the control center are all interconnected using an ad hoc network with multi-hop functionalities. In this swarm, followers act as relays, while the leader is always human piloted and responsible for real-time video capture.

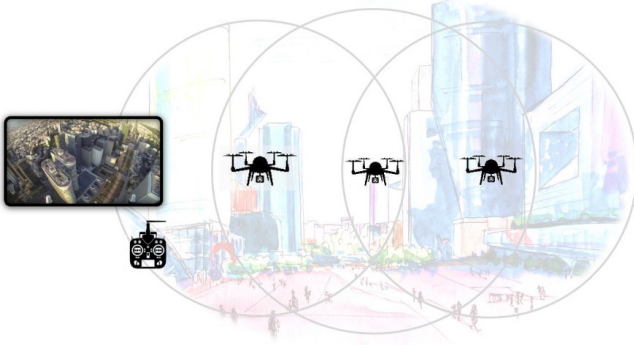


Fig. 1. A proposed ad hoc scenario with a basic line formation

By using our proposal, if the leader L is moved further (to capture a video of a very distant area, for instance), a first follower F_1 will automatically take off and position itself between the operator and the leader to relay the control commands, status information and video in real-time. If the leader goes still further (also moving F_1), a second follower F_2 will do the same, positioning itself between the operator and F_1 , and so on. Conversely, when the leader returns to the operator, the last drone that has taken off automatically will land if the signal quality is sufficient with the drone ahead. The prototype currently developed allows up to four followers with real time video streaming from the leader to the operator. Figure 1 shows a remote control with a screen (on the left) to command the drone leader (on the right) and view the captured real-time video from its camera. The distance between the command center and the leader being too long, a first follower, then a second one, automatically took off to ensure good video quality (depending on the requirement of the application).

Distances between drones are not predetermined. They are defined from the received signal strengths in dBm (circles in the figure), applying a smoothing function like the calculation of the RTT (round trip time) in TCP protocol. Indeed, drones are positioned according to the others, each one looking at the signal power received from the drone ahead. Unlike existing solutions where QoS, particularly in terms of throughput, is not taken into account, ours allows to determine the distance related to measured dBm value, according to the bit rate required by the application. It is important to emphasize that there is no need for any infrastructure or additional sensors to manage the swarm of drones or GPS modules. These are not useful since they are generally inaccurate and do not provide information about QoS.

B. System component

Figure 2 illustrates the basic components of our swarm to work as defined. Let L be the drone leader, F_i the i^{th} follower drone (F_1 being the first follower to take off when L is moving away), C the Control Center (e.g. a laptop or a Raspberry Pi) and $P_{x,y}$ the signal strength received by x from y . Only L is human piloted either by our software or by a mobile application. All the other drones are autonomous.

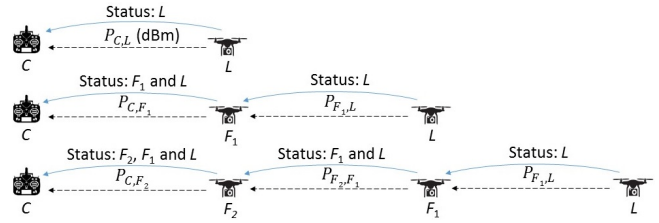


Fig. 2. Ad hoc communications between drones and control center

Our solution is based on two softwares, the former in the control center and the latter in every drone.

Control center A software module launched at C is intended to establish the ad-hoc network, discover the topology and manage followers' take-offs and landings. More precisely, once all the drones are connected, the leader can be controlled and it is labeled as the last drone, denoted l . We define two Quality Thresholds, T_{low} and T_{high} , to trigger a take-off or landing, respectively. The two following algorithms describe these two phases.

Algorithm 1: Automatic take-off algorithm

Data: An existing ad-hoc network between the drones and C , L already flying and controlled by the operator

Result: Automatic take-off of a follower

begin

if $P_{C,l} < T_{low}$ **then**
 Take-off a new drone F_i
 Assign label l to this new drone

repeat

| the condition
until no drone left

Therefore, T_{low} is the signal value needed to take off. We consider that $P_{C,l}$, the signal strength received by the control center from the last drone in the air, represents a good quality link if its value is higher than T_{low} . When $P_{C,l}$ passes under this minimum threshold, the degradation is too large to guarantee the requested quality, thus C sends a follower. In the above scenario, C measures periodically $P_{C,l}$. When a new follower is needed, C stops monitoring the signal strength from the last drone launched and starts measuring this from the new one.

Inversely, when the pilot directs the drone leader to return to its original position, all the flying swarm goes backward. When $P_{C,l}$ passes over the maximum threshold T_{high} , this means that the last follower is no longer useful because redundant to get the required quality. Once this follower, labeled l , is close enough to the pilot, C gives the order to land. The same process will be applied to all the followers in the formation if the drone leader comes closer and closer to C , as defined in Algorithm 2.

Algorithm 2: Automatic landing algorithm

Data: An existing ad-hoc network between the drones and C , L and at least a *follower* already flying

Result: Landing of a follower at initial position

```
begin
  if  $P_{C,l} > T_{high}$  then
    Land drone labeled  $l$ 
    Assign label  $l$  to the drone ahead
  repeat
    the condition
  until all the followers are on the ground
```

Drone software Another software is embedded in every drone, allowing each one to periodically send status messages to its follower (if any), forwarded to C . A drone receiving such a message treats the content and forwards it to the next drone in direction to C . Status messages are used to maintain information about the drones (e.g. which is followed, what is the height of the leader). The signal strengths are obtained from these. Let F_{i-1} be the drone followed by F_i , with $i > 1$. For F_1 , the followed drone is L . In other words, $F_0 = L$. After taking off, a drone must position itself correctly behind the drone followed, that is to say at the right distance. For this, we define two Movement Thresholds, T_f and T_b , to move forward or backward respectively. If the received signal strength is between these two values, the drone remains stationary, as indicated in the Algorithm 3 implemented in each drone (except the human piloted leader).

Algorithm 3: Automatic following algorithm in F_i

Data: An existing ad-hoc network between the drones and C , L and F_j , $1 \leq j \leq i$, already flying

Result: Automatic positioning to maintain required QoS

```
begin
  if  $P_{F_i, F_{i-1}} < T_f$  then
    Move forward
  else if  $P_{F_i, F_{i-1}} > T_b$  then
    Move backward
  else
    Stay still
  repeat
    the condition
  until as long as it is in the air
```

IV. IMPLEMENTATION

Our main interest has been to provide long distance real-time video surveillance using IEEE 802.11 in ad hoc mode. In Europe, transmission power is limited to 20 dBm (100 mw) giving about no more than several tens of meters for a single hop. Otherwise a set of high gain directive MIMO antenna is needed to reach further. But for several reasons, we have chosen to develop our solution with Bebop2 UAVs, which are commercial drones designed and marketed by Parrot [16]. Indeed, the Bebop2 has relatively small size (28x32 cm) and weight (500 g), with on-board ARM processor and flight time up to 25 minutes. Also, the drone instantly stops its engines when the propellers hit an object. In addition, Bebop2 has the

ability to stabilize in the air, even in an indoor environment thanks to a small optical flow camera. Finally, the drone works with a Wi-Fi antenna allowing the ad hoc mode and offers a SDK to take control through the embedded code.

A. Software Module

In order to carry out our test scenario, a distributed system was required. The software is implemented in C++ programming language using boost library [17]. We divided our software into two modules. The first one is a cross-compiled on-board module, within the Bebop2 firmware. It provides drone-autonomy and generates on-board control command based on received signal measured periodically by the module itself. The second module is off-board and it is installed in the control center. It is responsible for high level commands such as take-offs and landings. For this, it monitors in real-time the strength of the signal received from the last follower in the air to determine if a new one has to be launched or not. Concerning security, the module has also a failsafe feature that offers the possibility to take control of any drone in the formation in the case of drifting or system failure. Finally, drones also benefits from feature provided by the on-board modules, such as Status Messages. These messages are sent periodically and analyzed by the receiver, then forwarded to the control center.

B. Routing protocol

Even with the extensive maneuverability and high dynamic of drones, an existing routing protocol should be adapted to this configuration. As route changes depending on the formation, the system has a basic routing protocol designed specifically for this pattern (as described in figure 2). Each time a drone is introduced to the swarm, a new route is established between this drone and the last one. Each drone in the formation communicates with its follower, never with the control center (unless if it is the last one). As mentioned above, a drone communicates with its follower using Status Messages. These correspond to Hello messages used in standard ad-hoc routing protocol such as OLSR [18]. Generally speaking, Hello messages are used to detect the neighborhood, and transmit periodically to others information about known links and neighbors. Likewise, status messages are sent periodically to inform others about the formation, the altitude and current status of each drone.

V. EXPERIMENT & RESULTS

A. Basic line scenario

A real flight test has been conducted in outdoor environment using 3 Bebop2 drones. The scenario was that illustrated in Figure 2, where followers pursue the human piloted leader. The aim of this experiment is to validate the autonomy of the followers (in terms of positioning and displacement) using only the received signal strengths. By using logging feature provided by our software, the user is able to recover the speed of any drone on all three axes. Essentially in this test, we will

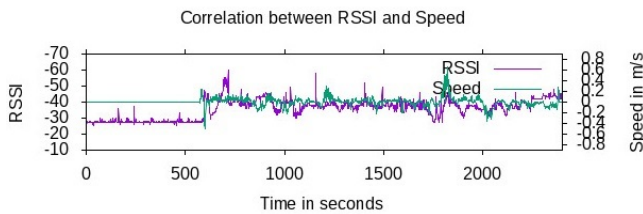


Fig. 3. Received signal power and speed of the first follower

consider a single axis, showing the horizontal displacement of the followers with respect to the power of the received signal (RSSI) by each of them, the leader moving only forward or backward in the experimentation performed. The following figure compares the speed of the first follower in real-time compared to the signal strength received from the leader.

As we can see, at the beginning of the experimentation, the leader has started to fly. However, it is too close from the control center to launch a follower. After that, the leader is moved away, thus the RSSI measured by the control center becomes below the fixed quality threshold. Therefore, the follower takes off. Apart from rare non-standard values, we can see that the RSSI remains between the two thresholds fixed in our experiment, the follower moving as it should to maintain this signal quality. Results obtained for next follower are comparable to the previous ones.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a solution using only ad hoc communications to synchronize and orchestrate a swarm of drones. No pre-flight infrastructure nor specific sensor is needed. Drones position relative to others based on the signal strength received from those ahead. Two softwares have been developed, one for the control center to manage followers' takes-off and landings, the other to be embarked in every follower for autonomous positioning. After presenting the general idea of the algorithms used, we have implemented our code in several drones and conducted experiments. The results show the correlation between the received signal power and the displacement speed of an autonomous drone.

Future work will focus on finding a more generic algorithm that will allow to carry out more sophisticated formations without the need for an existing coordination system. For this, we might also include values obtained from the direct use of IMU (Inertial Measurement Unit) in order to construct other patterns, such as triangles or hexagons. An adaptation of the existing ad-hoc routing protocol could be relevant to integrate safety and security measures (e.g. change or removal of a drone on the fly). In addition, instead of using the current form of dynamic routing that depends on the topology, others (like QOLSR [20]) should benefit from quality of service features. Finally, the first experiments conducted in an indoor environment have shown good performance, but it is necessary to adjust the smoothing function of the signal strength. This function will be made dynamic to adapt automatically to the environment.

REFERENCES

- [1] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3472–3479.
- [2] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 45–52.
- [3] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D'Andrea, "Building tensile structures with flying machines," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3487–3492.
- [4] Verity Studios. [Online]. Available: <http://veritystudios.com/>
- [5] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957415813002262>
- [6] M. Mller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 5113–5120.
- [7] R. Ritz, M. W. Mller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4972–4978.
- [8] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 56–65, Sept 2010.
- [9] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Auton. Robots*, vol. 35, no. 4, pp. 287–300, Nov. 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10514-013-9349-9>
- [10] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 3131–3137.
- [11] P. Chandhar, D. Danev, and E. G. Larsson, "Massive mimo as enabler for communications with drone swarms," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 347–354.
- [12] G. Loianno, J. Thomas, and V. Kumar, "Cooperative localization and mapping of mavs using rgb-d sensors," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4021–4028.
- [13] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1736–1741.
- [14] D. Panagou and V. Kumar, "Cooperative visibility maintenance for leader x2013;follower formations in obstacle environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 831–844, Aug 2014.
- [15] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots*, Apr. 2016.
- [16] PARROT. [Online]. Available: <http://www.parrot.com>
- [17] Boost C++ Libraries. [Online]. Available: <http://www.boost.org/>
- [18] OLSR. [Online]. Available: <http://www.olsr.org>
- [19] "802.11-2012 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE-Inst, Tech. Rep. IEEE Std 802.11-2012. [Online]. Available: <http://ieeexplore.ieee.org/servlet/opac?punumber=6178209>
- [20] H. Badis, "Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR)," Internet Engineering Task Force, Internet-Draft draft-badis-manet-qolsr-05, Mar. 2007.