Ventures, Insights, and Ponderings from Real-World experiments with FANETs and UAVs

Bruno José Olivieri de Souza[1], Markus Endler[1]

***Abstract*—In the research on FANETs (Flying Ad-Hoc Networks) and distributed coordination of UAVs (Unmanned Aerial Vehicles), also known as drones, there are many studies that validate their proposals through simulations. Simulations are important, but beyond them, there is also a need for real-world tests to validate the proposals and enhance results. However, field experiments involving drones and FANETs are not trivial, and this work aims to share experiences and results obtained during the construction of a testbed actively used in comparing simulations and field tests.**

I. Introduction

In a significant subset of academic works in computer science, proposals, when not formally proven, undergo extensive verification through testing and analysis. These tests yield results from datasets previously available in knowledge areas such as Database and Supervised Learning, or primarily from experiments planned and implemented in environments simulated by the proponents themselves.

Simulation-based analyses and verifications are common across diverse fields like networks, distributed systems, computer vision, and optimization. Simulations offer accessibility and can be conducted by individual researchers, facilitating faster research progress without the need for expensive equipment, which may not be reusable for other studies. Simulations accelerate and facilitate scenarios that were previously inaccessible, but they simplify all environmental variables affecting an experiment. This simplification increases notably when researchers simulate using their own implementations instead of using robust frameworks like NS3, OMNET++/INET, especially in proposals related to computer networks.

In addition to verifications, validations in real-world scenarios can also be performed. In this way, numerous environmental factors, scenarios, and equipment start to influence the experiments of a proposal under analysis. These validations, or realistic experiments[1], enable significant or total realism and coupling with reality, making the results more accurate.

This work contributes with some experiences in the construction, usage, and maintenance of a testbed designed for validations of proposals and works that involve issues related to networks and distributed systems in the context of FANETs (Flying Ad-Hoc Networks). This testbed, called GrADyS Framework, is composed of simulators, autonomous UAVs (drones), as well as sensor networks effectively deployed in the field, along with other equipment such as autonomous rovers (UGVs) that move on the ground. Some field measurement results are also presented in contrast to simulations and their implications. The GrADyS project aims to perform simulated verifications and field validations to enhance discussions on swarm drone coordination [1].

II. GrADyS Framework

The GrADyS Framework comprises the integration of various components, with its greatest strength lying in the ability to validate simulations conducted with drones and, in some cases, wireless sensor networks (WSNs) in the real world. Figure 1 illustrates the project's working approach.

The framework can be observed through its four pillars: (1) Simulation of protocols and distributed algorithms for coordination of mobile nodes with ad hoc communication; (2) Simulations of WSNs and routing algorithms; (3) Coordination of UAVs in real flights; and (4) The effective implementation of a WSN with ground sensors.

The simulation of protocols and distributed algorithms for the coordination of mobile nodes is enabled by using the same coordination code written in Python, designed to be executed in a simplified manner for rapid prototyping in simulations and vehicles. Simultaneously, the same work can make use of the OMNET++/INET suite to execute communications more faithfully and coordinate the movement dynamics of aerial vehicles in a Software-In-The-Loop (SITL) simulator for real flight controllers based on Ardupilot. Specific results in this line of work can be analyzed in the GrADyS-SIM project[2], based on OMNET++/INET, MAVSIMNET[2] which facilitates the integration between the Mavlink protocol and OMNET++/INET, as well as various other aspects available on the project's page[3] and its blog[4].

The simulations of WSNs and routing algorithms are implemented by inheriting the implementation stacks present in OMNET++/INET, where BLE and 802.15.4 implementations have yielded results in previous works[3].

The actual implementation of WSNs on the ground may be one of the scarcest validations in related works. In our research, we modified the native implementation stack of controllers based on the ESP32 chip to provide BLE routing algorithms with more energy-efficient use than the BLE specification itself[4].

The coordination of UAVs in real flights is conducted using low-cost, low-sophistication quadcopters, whose parts are individually acquired and assembled by students. The vehicles are equipped with single-board computers (SBC),

[1]DSc Bruno Olivieri and Dr. rer. nat. Markus Endler at PUC-Rio (bolivieri|endler)@inf.puc-rio.br

[1]https://youtu.be/rz02rnxYYQY?si=hISwZ3T3D89Gy1dZ

[2]https://thlamz.github.io/MAVSIMNET/

[3]https://www.lac.inf.puc-rio.br/index.php/gradys/

[4]https://gradys.tumblr.com/

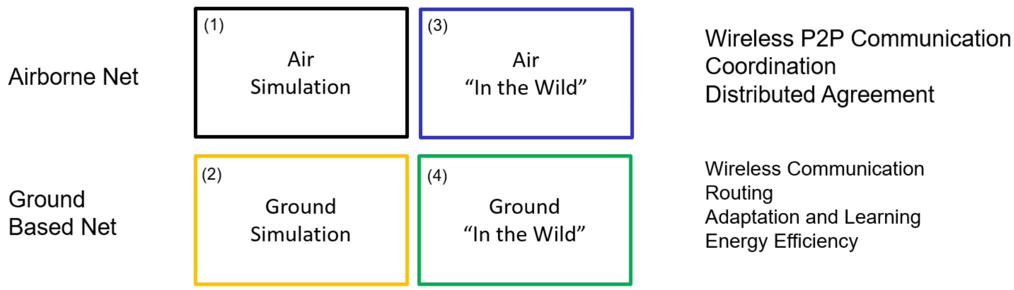


Figure 1: The methodology of the GrADyS Project, illustrating its 4 simulation components in the laboratory and in the field (a.k.a. in the wild).

typically Raspberry Pis, enabling the coding of protocols and distributed coordination algorithms while having 802.11 and 802.15.4 radios. Verification with real vehicles enables a whole new level of confidence in the tests while presenting new challenges for test execution.

In the real-world implementation, the framework is realized through various components illustrated in Figure 2. It showcases both simulation components and field testing components.

III. Related Work

In swarm UAV coordination protocol research, several studies implement simulators to present and verify their findings. Notable works include AirSim[5], FlyNetSim[6], GrADyS-SIM[2], ArduSim[7], and many others.

The vast majority of studies focus on simulation, and typically, the portion related to communication between nodes is less reliable and simplified. Exceptions to this statement, to the best of our knowledge, are present when there is synergy with ROS environments and in MAVSIMNET.

The implementation of reusable real-world testbeds for the validation of proposals with UAVs has been around for just over a decade, starting with the work of Vijay Kumar et al[8] at the University of Pennsylvania. In this work, various micro UAVs communicate with a central node that controls them with a centralized Matlab model. The entire positioning system is indoor-enabled with image capture systems[9], such as VICON[5]. Following this work, other reusable testbeds with the same architecture were implemented by academia, such as at ETH by Rafaello D'Andrea et al in 2014[10], or even with underwater vehicles, as in the work of Sidney et al.[11].

Over a decade later, new works are presented with a similar architecture, as in the case of Guerreiro et al.[12], where the notable difference is that the positioning system of UAVs is controlled by UWB (ultra-wideband), and control remains centralized, emulating independence between nodes and issues in FANETs.

The mentioned cases often lack outdoor tests, which pose greater challenges due to interferences, including environmental, telecommunications, and logistical factors. Additionally, unlike other works where processing occurs on a central node, our work tests distributed algorithms on separate nodes communicating in a fully Ad Hoc FANET.

IV. Lessons Learned / Our Real-World Experiences

Even in more mature research groups or thematic laboratories with multiple researchers in robust university departments, research often occurs compartmentalized or even individualized. Around the same theme, several postgraduates and young researchers conduct their research and implement their simulations individually. This becomes practical in terms of verifications. However, it may hinder the reuse and reproducibility of research.

When we start talking about validations, this scenario encounters two main obstacles: the first is that the costs associated with transitioning from simulations to real-world testing almost always lead research teams to question how to reuse the investments made. Secondly, a single researcher is unlikely to be able to implement and execute real-world experiments with various equipment, as shown in Figure 3[6].

A. *"Leaving the laboratory"*

Acquiring equipment and setting up infrastructure for experiments can be daunting. Planning and funding are time-consuming, further complicated by insufficient support. In the GrADyS testbed described in Section II, deploying drones with multiple radios requires meticulous logistics. Besides drones, essential supporting equipment and space allocation are crucial, from limited campus areas to vast, remote expanses. Space constraints and logistical complexities force compromises, impacting the feasibility and efficiency of outdoor tests. Challenges include adverse weather and ensuring equipment usability. Basic infrastructure is indispensable, with accessories like umbrellas and mobile tables becoming essential tools. Despite challenges, the GrADyS project perseveres, overcoming hurdles to advance research in drone and WSN technologies. Several images from these moments can be accessed at https://gradys.tumblr.com/.

[5]https://www.vicon.com/

[6]Video at https://youtu.be/VCO6h4jAAQk?si=SzdE1psVXQYvac-m

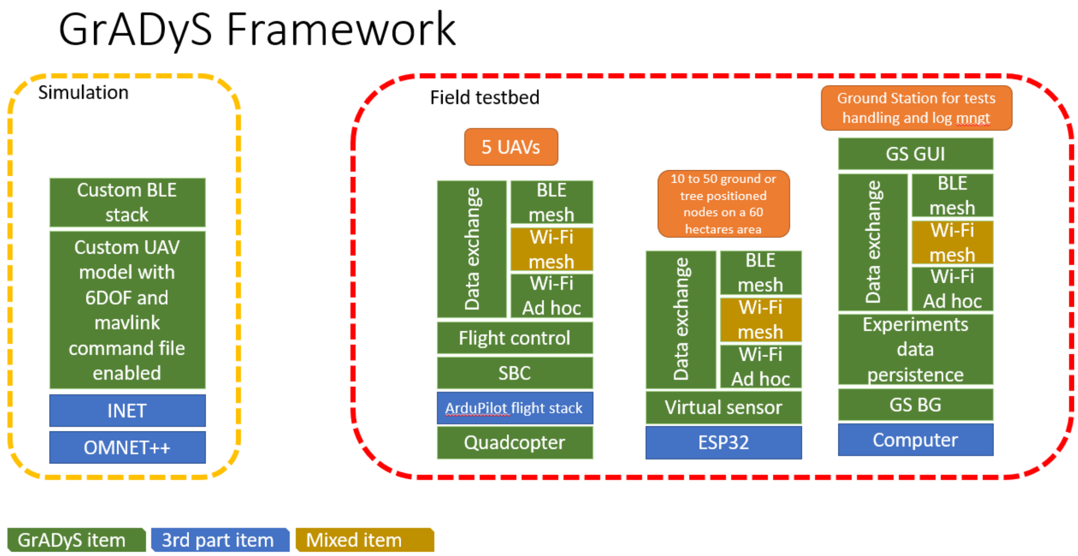


Figure 2: The GrADyS framework illustrating its simulation components in the laboratory and its field components.

Figure 3: Four drones taking off synchronously.

B. Vehicle in a FANET

Vehicles, by their nature, are included in experiments to move, and drones take this to a 3D environment. Outdoors, they are quite susceptible to winds and dangerous obstacles. Preparing the vehicles takes time, and at least one person per vehicle is required. If it takes ten minutes to enable one drone and there are six drones, either six people are needed, or it takes over an hour.

The decision on whether to acquire ready-made drones or assemble custom prototypes can be difficult. Taking the example of real-world tests for a FANET[13], the drones are only the mobile nodes that are not part of the main research contribution, but they are extremely necessary. Furthermore, a research group aiming for scientific contributions in FANETs may not have the know-how for assembling, maintaining, and piloting, or even automating drones.

These factors usually contribute to the acquisition of consumer-grade equipment, ready for use, and in this market, the DJI brand dominates. These devices have a very low entry barrier and, initially, appear to be the most agile choice. However, they are not designed to carry any payload or enable an SDK that allows for significant customization or automation.

Before deciding which drones or other vehicles will be used, a deeper analysis is always important so that expectations beyond the initial ones can be understood, and the investment can be reused or long-lasting. The DJI brand remains an excellent option as long as it is within a family of drones that integrates with ROS[7]. DIY vehicles with OpenHardware and OpenSoftware projects will inevitably have a longer maturation time but will be widely reusable and customizable for experiments. In this latter line, the ArduPilot software suite is robust and caters to virtually any type of vehicle with a standardized set of APIs.

For the use of drones, logistics before, during, and after flights cannot be underestimated, as a lot of time and effort are spent. Several activities need to be conducted simultaneously, from piloting or monitoring autonomous flight safely to controlling the test area to ensure safety and observing data collection. A simple error in data collection detected after field activities can result in the loss of all efforts, requiring the entire process to be repeated.

C. Batteries

Drones use batteries invariably crafted with Lithium. These batteries have the advantage of high discharge power. However, these characteristics make the batteries challenging to acquire directly, as there are various transportation and storage restrictions, translating into high costs.

A battery can be charged in up to an hour at a 1C charging rate and should provide a flight time of ten or, at most, 20 minutes. This brings two more challenges: (1) many batteries are needed for a good day of testing, and (2) for the batteries to be charged in a reasonable time, parallel charging is necessary.

[7]The Robot Operating System - https://www.ros.org/

A challenging point is that Lithium batteries should not be stored fully charged or fully discharged.

These batteries are hazardous as they can catch fire and even explode. Improper handling of LiPo batteries can lead to fire, explosions, and inhalation of toxic smoke. The use of batteries must be planned with caution, and storage recommendations must be strictly followed. Activities for charging the batteries should always be observed.

D. Radios

Once the obstacles related to FANET vehicles are overcome, radios become an important point. Among various chip and device options, it is possible to create two broad groups. First, with a strong connection to microcomputers, there is a multitude of radios that enable 802.11 and, in turn, IP networks. Second, there are radios with lighter and similar stacks such as 802.15.4, LoRa, ESP-NOW, among others[14].

For applications that exchange structured or voluminous data, it is interesting to use radios that support some version of 802.11. For small exchanges of integers or minimal data, radios 802.15.14 should be prioritized. Although these statements may seem straightforward based on the nature of the two mentioned standards, they come from experience in using the radios that support them. The stacks and APIs of both cases are diametrically opposed in the offerings of functionalities and usage.

On the one hand, using the 802.11 standard provides almost countless interoperable equipment and implementation libraries. On the other hand, the 802.15.4 standard has a scarce supply of libraries, and its sub-standards (LoRa, ESP-NOW, ZigBee, etc.) fragment the hardware offering and greatly restrict interoperability. Radios based on the 802.15.4 standard will allow much simpler communication with greater range than radios delivering the 802.11 standard, and this should be considered.

After discussing radios, the use of antennas deserves special attention. Many devices come with antennas printed directly on their boards, such as some ESP32 and the Raspberry Pi. These antennas usually work well at short distances, but when these devices are installed on drones, the range is restricted to a few meters, making any tests unfeasible. It is worth noting that the drones' motors generate a lot of noise in the circuit, and low-pass filters should be used.

External antennas separate from the equipment and of good quality should always be prioritized. The market is flooded with antennas cut incorrectly for the desired wavelength. In our testbed, drones using Raspberry Pi with the original antenna were accessible at a maximum of 10 meters when in flight. Using an external antenna and USB radio, we were able to reach 40 meters. Finally, the original radio of the Raspberry Pi, with its standard antenna removed and a new modified antenna installed, reached a range of 200 meters, the maximum expected for the 802.11 standard.

Another important point about antennas that cannot be underestimated is their positioning. It is expected, and we have tested, that two simple ESP32s using 802.15.4 can exchange data at 180 meters when their antennas are parallel to each other, perpendicular to the ground, and 2 meters above the ground. If the antennas are laid near the ground at about 10cm, this range of 180 meters is reduced to 40cm. This is entirely expected given the signal's radiation.

E. Companion Computers

For experiments involving distributed systems, the nodes must be logically independent, and in the case of swarms of drones running effectively distributed algorithms, embedded processing nodes are required. These nodes should have the aforementioned radios and antennas. Considering that the drone will be responsible only for flight control processing, the experiment must be executed on the companion computer that will be embedded in the drone. To fulfill this role of a companion computer, two options are usually considered: (1) System on a Chip (SoC) such as ESP32 and others; (2) Single Board Computers (SBC) such as Raspberry Pi, NVidia Xavier, and others. Both have advantages and disadvantages:

i SoC:
- Advantages: They are very small and easy to embed, lightweight, and inexpensive. They are also more durable than SBCs and can work for a long time on batteries;
- Disadvantages: Limited processing power; it's important to note that the logical processing cores for the inserted code may be the same as those processing radio packets, leading to potential interference. Some programming paradigms are not available, such as multiprocessing. The code is usually manually inserted ('flashing' the firmware), making it impractical and challenging to customize. There is limited reuse of other open-source research and projects.

ii SBC:
- Advantages: Good processing power, ample memory, multiple embedded radios (at least WiFi and BLE), highly expandable, use of operating systems, and other possibilities similar to a complete computer, such as a laptop. There is a lot of reuse from other open-source research and projects;
- Disadvantages: They are more expensive. In fact, a Raspberry Pi 4 with 4GB currently costs the same as a mini PC with an Intel N5095 processor, which has many more resources but is not easily embeddable. SBCs do not work as long on batteries as SoCs and are not always small and light enough to be embedded.

iii Tradeoffs: Although SoCs enable a quick start and "quick wins", they are very scarce in resources. This becomes more challenging to manage outside the core of the experiment, in solutions that run in parallel, collecting data, storing logs, and organizing them. In our experiences, even when focusing on a Nordic SoC[8]

[8]https://www.nordicsemi.com/products/nrf52840

or ESP32, it was inevitably necessary to use an SBC between them and the drones to manage the experiment, collect data, and control the drone. Moreover, the use of an embedded SBC instantly enables a large set of tools on top of the operating system, such as Python, ROS, ROS2, databases, and messaging brokers.

F. Logs and synchronization

One issue that consumes a lot of time in network and distributed systems research is debugging code that often runs in some form of asynchrony or parallelism. To address this point, many IDEs provide features to observe values in parallel, and developers generate a significant amount of logs.

However, in real-world experiments, there is a significant complicating factor that does not occur in simulations. SoCs and SBCs invariably lack an onboard Real-Time Clock (RTC) with a battery. Even if memory cards are used for logging mechanisms, there will still be the difficulty of synchronizing them. SBCs usually synchronize their clocks via the Internet, and in a FANET, this may not be available. RTCs can be manually installed, but they are generally not very accurate and require more customization in the drone's electronics.

One solution we adopted was to access the drones' APIs to retrieve the time from the GPS, which is extremely accurate. Although this solution may seem trivial, it is laborious.

In all cases, even with synchronized logs on the nodes, they are spread across each node, each drone, or sensor. Retrieving them can be as time-consuming as the battery charging process. For this, a Ground Station (GS) is useful and can collect data in the field for a check if they are intact, and the experiment can indeed be analyzed outside the field. Managing multiple SD cards and files is daunting.

G. Ground Stations

In the end, every experiment involves data collection. In our past experiences, each student typically started independently, building a pair of scripts that resulted in a comprehensive set of code and analyses. This was the case when they didn't evolve into complete Ground Stations (GS) to control the drones and experiments.

This effort can be better utilized if planned and reused across multiple research lines. In this regard, the common goal is often to observe what is happening, collect data, and send commands to a subset or all drones.

With this in mind, our project implemented and provided GrADyS-GS[15][9]. It is a web-based GS that is easy to maintain and customize, implemented in Python and JS, focusing on the simple exchange and storage of JSONs between vehicles and the GS. With it, multiple people can observe the same experiment on various terminals and simultaneously control parts of it collaboratively.

[9]https://github.com/Project-GrADyS/gradys-gs

V. Some results and lessons learned

Continuing the path of Validating our proposals and simulations, we present three experiments and some lessons learned. In all our experiments, we explore paradigms without centralized infrastructure for communications, with the GS being only an Ad Hoc node, but stationary.

A. Mesh Networks - 802.11s

The test involved flying drones over a field, collecting data from 10 ground sensors, and quantifying them. In these experiments, each drone used a RaspBerry Pi solely for drone control and an ESP32 as a radio and for running the test itself. On the ground, each sensor was equipped with an ESP32, as shown in blog. A straightforward test to evaluate the components of the environment.

Communication between the nodes utilized ESP32 radios as a mesh network, employing PainlessMesh[10] with the radio configured to 11 dBm. This setup aimed to exchange JSON data between drones and nodes on the ground.

Figure 4a illustrates the results of one of the experiments. Vertically arranged are sensors from S1 to S10. Horizontally, three bars represent message exchanges at three different drone flight altitudes: 20 meters, 35 meters, and 50 meters. All flights were conducted at 5 m/s.

Initially, one would expect that the closer two radios were, the more data they could exchange given the signal strength. Conversely, a drone flying higher would encounter fewer obstacles between it and another node. In this case, with a better line of sight between radios, more data would be exchanged between nodes.

The interesting point in this simple experiment was observing that one node never exchanged data with the others, and at some altitudes, there was never any data exchange. Further analysis over several days of field testing revealed that the nodes couldn't establish a link between them. This was challenging with drones flying at a relatively slow speed of 5 m/s, and there was never significant data exchange even with flights at 10 m/s in this specific scenario. It's worth noting that one node never exchanged data, even when not in a significant shadow zone.

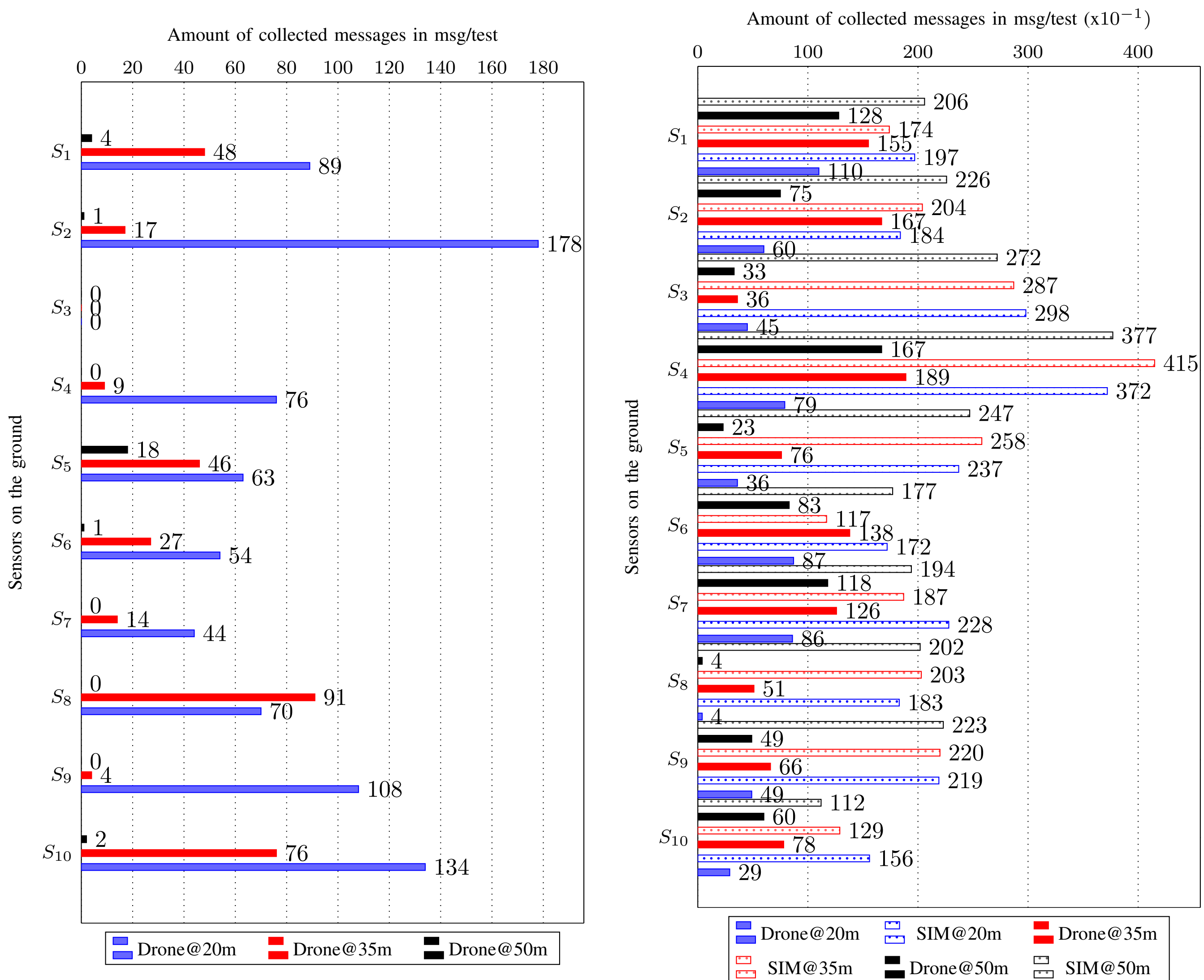
The entire software setup to enable a mesh network often required more time than the handover between nodes. While an infrastructure-based network might be impractical in various scenarios, a mesh network might not be a viable alternative either. In fact, there are studies proposing specific MAC protocols for aircraft rendezvous[16].

B. Broadcast-based Networks - 802.15.4.

Another approach for the experiment presented in Section V-A was to replace the use of 802.11s with the implementation of a 802.15.4 stack. In this case, ESP-NOW[11] was employed, enabling a broadcast-oriented paradigm with payloads of up to 250 bytes.

[10]https://gitlab.com/painlessMesh/painlessMesh

[11]https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html



(a) 802.11s

(b) 802.15.4

Figure 4: Test of WSN/FANETs with 802.11s and 802.15.4

The same map, locations, and drones were used, and an identical environment was also simulated. Figure 4b presents the results. In this case, all nodes managed to exchange data on a scale ten times larger. Besides the data exchange using real drones, the results of simulations (SIM@) performed in GrADyS-SIM[2] in OMNET++/INET are also presented. Further details on how these results were used to enhance the simulation can be found in Olivieri et al.'s work[1].

C. 802.11 Ad Hoc

Continuing our effort to enable FANETs for coordinating swarms of drones, we shifted towards using Single Board Computers (SBC) running Linux embedded in the drones. In these SBCs (Raspberry Pi 4), the traces connecting the radios to the printed antennas on the boards were removed, and ipex connectors were soldered[12]. This allowed external antennas to be installed on the SBC, using high-quality WiFi router antennas.

An 802.11 Ad Hoc network was configured on the Linux-running SBCs, with the flexibility to adjust their device drivers when necessary. Consequently, a network with IP support was enabled. The experiment aimed to test the traffic between two drones at incremental distances. One drone remained stationary at 10 meters in height, while the other moved away in 20-meter intervals, reaching a distance of 200 meters. At each stop, the moving drone conducted tests with the default parameters of the iPerf3 tool[13]. Figure 5 illustrates the results. The initial observation focused on transfer rates using TCP. As expected, transfer rates decreased as the drones moved farther

[12]https://youtu.be/MTwWnZG8wUY?si=MF3wNg9aAs3Mc5mi
[13]https://iperf.fr/

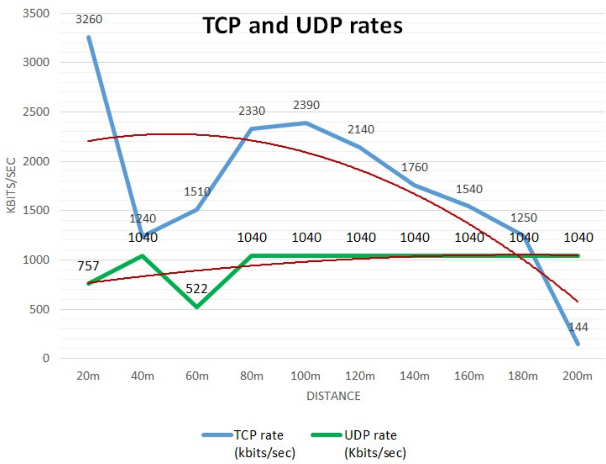


Figure 5: Measuring the bandwidth between two drones.

apart. If the tested application does not require the exchange of voluminous data such as images or sound, this may not pose a problem. However, for applications demanding greater link robustness between drones, the use of UDP with flow control may be more resilient as packet loss increases. By regulating the UDP transmission rate to 1 Mbit/s, a stable link could be maintained up to 200 meters.

VI. Conclusions

Simulations are fundamental and significantly expedite the research process, whether in the context of FANETs or any proposals aligned with distributed systems. However, real-world tests are indispensable for validating certain propositions. When testing hypotheses in the field, variables from the environment, absent in simulations, come into play. With these new variables, the hypothesis undergoes a more robust test, and simulators can be significantly improved, as mentioned in the reference article in Section V-B. For real-world tests, no preparation can be underestimated. More personnel can improve field test control. The tradeoffs of equipment choices must be thoroughly analyzed, and reusability and customization can be emphasized.

This study was financed in part by AFOSR grant FA9550-23-1-0136.

References

[1] B. J. O. De Souza, T. Lamenza, M. Paulon, V. B. Rodrigues, V. C. A. Carneiro, and M. Endler, "Collecting Sensor Data from WSNs on the Ground by UAVs: Assessing Mismatches from Real-World Experiments and Their Corresponding Simulations," *Proceedings - IEEE Symposium on Computers and Communications*, vol. 2023-July, pp. 284–290, 2023.

[2] T. Lamenza, M. Paulon, B. Perricone, B. Olivieri, and M. Endler, "GrADyS-SIM – A OMNET++/INET simulation framework for Internet of Flying things," in *SBRC 2022 - Brazilian Symposium on Computer Networks and Distributed Systems*, Fortaleza, 2022. [Online]. Available: https://sbrc2022.sbc.org.br/salao-de-ferramentas/

[3] M. Paulon J.V., B. J. Olivieri de Souza, and M. Endler, "Exploring data collection on Bluetooth Mesh networks," *Ad Hoc Networks*, vol. 130, no. February, p. 102809, 2022. [Online]. Available: https://doi.org/10.1016/j.adhoc.2022.102809

[4] M. Paulon, B. J. Olivieri De Souza, T. De, S. Lamenza, and M. Endler, "Practical Challenges And Pitfalls Of Bluetooth Mesh Data Collection Experiments With Esp-32 Microcontrollers," Nov 2022. [Online]. Available: https://arxiv.org/abs/2211.10696v1

[5] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," *Springer Proceedings in Advanced Robotics*, vol. 5, pp. 621–635, 2018. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-67361-5_40

[6] S. Baidya, Z. Shaikh, and M. Levorato, "FlynetSim: An open source synchronized UAV network simulator based on ns-3 and ardupilot," *MSWiM 2018 - Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 37–45, Oct 2018. [Online]. Available: https://doi.org/10.1145/3242102.3242118

[7] F. Fabra, C. T. Calafate, J. C. Cano, and P. Manzoni, "ArduSim: Accurate and real-time multicopter simulation," *Simulation Modelling Practice and Theory*, vol. 87, pp. 170–190, Sep 2018.

[8] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro-UAV testbed," *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, Sep 2010.

[9] A. I. Hentati, L. Krichen, M. Fourati, and L. C. Fourati, "Simulation Tools, Environments and Frameworks for UAV Systems Performance Analysis," *2018 14th International Wireless Communications and Mobile Computing Conference, IWCMC 2018*, pp. 1495–1500, Aug 2018.

[10] 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*, 2014.

[11] N. Sydney, S. Napora, and D. A. Paley, "A multi-vehicle testbed for underwater motion coordination," in *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop*, ser. PerMIS '10. New York, NY, USA: ACM, 2010, pp. 107–111. [Online]. Available: http://doi.acm.org/10.1145/2377576.2377597

[12] H. Cabrita and B. Guerreiro, "NOVA.DroneArena: design and control of a low-cost drone testbed," *Proceedings - 2021 International Young Engineers Forum in Electrical and Computer Engineering, YEF-ECE 2021*, pp. 20–25, 2021.

[13] I. Bekmezci, I. Sen, and E. Erkalkan, "Flying ad hoc networks (FANET) test bed implementation," *RAST 2015 - Proceedings of 7th International Conference on Recent Advances in Space Technologies*, pp. 665–668, Aug 2015.

[14] L. P. Fraile, S. Tsampas, G. Mylonas, and D. Amaxilatis, "A Comparative Study of LoRa and IEEE 802.15.4-Based IoT Deployments Inside School Buildings," *IEEE Access*, vol. 8, pp. 160 957–160 981, 2020.

[15] B. Perricone, T. Lamenza, M. Paulon, B. J. O. de Souza, and M. Endler, "GrADyS-GS – A ground station for managing field experiments with Autonomous Vehicles and Wireless Sensor Networks," *arXiv*, Apr 2022. [Online]. Available: https://arxiv.org/abs/2204.00488v1

[16] M. A. Araghizadeh, P. Teymoori, N. Yazdani, and S. Safari, "An efficient medium access control protocol for WSN-UAV," *Ad Hoc Networks*, vol. 52, pp. 146–159, Dec 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S157087051630213X?_rdoc=1&_fmt=high&_origin=gateway&_docanchor=&md5=b8429449ccfc9c30159a5f9aeaa92ffb&dgcid=raven_sd_recommender_email&ccp=y