


## Article

# Inter-UAV Routing Scheme Testbeds

Georgios Amponis <sup>1</sup>, Thomas Lagkas <sup>1,\*</sup> , Panagiotis Sarigiannidis <sup>2</sup> , Vasileios Vitsas <sup>3</sup> and Panagiotis Fouliras <sup>4</sup><sup>1</sup> Department of Computer Science, International Hellenic University, 65404 Kavala, Greece; gamponis@ihu.gr<sup>2</sup> Department of Electrical and Computer Engineering, University of Western Macedonia, 50100 Kozani, Greece; psarigiannidis@uowm.gr<sup>3</sup> Department of Information and Electronic Engineering, International Hellenic University, 57400 Thessaloniki, Greece; vitsas@ihu.gr<sup>4</sup> Department of Applied Informatics, University of Macedonia, 54636 Thessaloniki, Greece; pfoul@uom.edu.gr

\* Correspondence: tlagkas@cs.ihu.gr

**Abstract:** With the development of more advanced and efficient control algorithms and communication architectures, UAVs and networks thereof (swarms) now find applications in nearly all possible environments and scenarios. There exist numerous schemes which accommodate routing for such networks, many of which are specifically designed for distinct use-cases. Validation and evaluation of routing schemes is implemented for the most part using simulation software. This approach is however incapable of considering real-life noise, radio propagation models, channel bit error rate and signal-to-noise ratio. Most importantly, existing frameworks or simulation software cannot sense physical-layer related information regarding power consumption which an increasing number of routing protocols utilize as a metric. The work presented in this paper contributes to the analysis of already existing routing scheme evaluation frameworks and testbeds and proposes an efficient, universal and standardized hardware testbed. Additionally, three interface modes aimed at evaluation under different scenarios are provided.

**Keywords:** FANETs; ad hoc networking; drone swarms; efficient routing



**Citation:** Amponis, G.; Lagkas, T.; Sarigiannidis, P.; Vitsas, V.; Fouliras, P. Inter-UAV Routing Scheme Testbeds. *Drones* **2021**, *5*, 2. <https://dx.doi.org/10.3390/drones5010002>

Received: 30 November 2020

Accepted: 23 December 2020

Published: 28 December 2020

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The birth of Mobile Ad hoc Networks (MANETs) has enabled the decentralization of a massive amount of applications and services. Ad hoc networks can now function in a great variety of environments and conditions without necessarily interfacing with the rest of the internet. All cooperating nodes in such networks are considered peers, with a master–slave relationship between them being nonexistent due to the networks fundamental architecture: each node is both an end-device and a router. Vehicular Ad hoc Networks (VANETs) were the first type of MANETs to be deployed, followed by Flying Ad hoc Networks (FANETs) which are the main focus of this paper. Another type of ad hoc networks which present great interest are cooperative VANET-FANET networks, in which UAVs function essentially as relays between vehicles. According to [1,2], FANETs differ from VANETs and other types of (ad hoc) networks in matters of:

- Node mobility (in terms of Velocity / Acceleration / Jerk)
- Topology (FANETs may function in a 3D grid in addition to the more traditional single-layered swarm topology [3]; additionally, a UAV-comprised network shall alter its topology with a significantly higher frequency, due to its nature)
- Availability of energy (arguably the most significant constrain of autonomy in an ad hoc network)

FANETs find application in both military and civilian fields, such as the ones described in [4–15]:

- Real time monitoring and surveillance
- Search-and-rescue operations

- Agriculture
- Relaying of communication

As envisioned in [16], while the challenges of swarm formation are numerous (routing, path planning, QoS provision, UAV coordination, etc.), multi-UAV deployments are the future of aerial communications. In surveillance and monitoring applications, routing of information is a means to an end; such is the case in search and rescue-oriented FANET deployments as well as the ones centered around agriculture. High route selection efficiency is mandatory for an acceptable quality of service (QoS) in all mentioned use-cases. As a general rule, routed data shall incorporate desired payload and control messages which forces the scheme developer to take into account the desired application class and packet overhead alike. When considering the relaying of communication between distant parties, routing (and high efficiency thereof) becomes not just a means to an end but rather a goal in itself. Such a use-case, centered around communication relaying as well as VANET-to-FANET cooperation is mentioned in [17]; the researchers have even proposed an application-specific cross-layer routing protocol aimed at enabling close cooperation between VANETs and FANETs, as mentioned previously. The importance of routing efficiency becomes even more highlighted when considering military or emergency/first response operations [4].

There exist a vast number of routing protocols, with many being developed for a very specific use-case and others functioning as a foundation for other schemes. Evaluation of existing/under development protocols is done traditionally using classical simulation software (more often than not, developed for non-ad hoc networks). To counteract the limitations arising from a such approach and obtain usable results, developers resort to the addition of mobile properties to the individual nodes comprising their networks. Thus, in most cases, developed routing schemes are evaluated using conventional networking simulators, which can in many cases, efficiently approximate a mobile ad-hoc behavior. Different mobility models are included as libraries for discrete event network simulators. Support for the following mobility models is inherent in the NS-3 network simulator:

- Constant Position Mobility Model (NS-3 specific mobility model)
- Constant Velocity Mobility Model (NS-3 specific mobility model)
- Constant Acceleration Mobility Model (NS-3 specific mobility model)
- Gauss Markov Mobility Model [18,19]
- Hierarchical Mobility Model (NS-3 specific mobility model; a combination of specified “parent” and “child” mobility models)
- 2D Random Direction Mobility Model [19]
- 2D Random Walk Mobility Model [20]
- Random Waypoint Mobility Model [21]
- Steady State Random Waypoint Mobility Model (NS-3 specific mobility model; uses stationary instead of random distribution for initial speed, pause and position)
- Waypoint Mobility Model (NS-3 specific mobility model)

For the majority of the existing routing protocols, a discrete event network simulator such as the NS series may prove sufficient. However, non-classically layered schemes cannot be as effectively simulated due to inherent limitations of said software. To address this issue, there was developed the FINS framework capable of considering flexibility in the layer stack and thus enable nonadjacent interlayer data exchange during simulation [22–24]. Opnet Modeler has also been used by researchers to evaluate routing schemes of both classical and cross-layer architectures in numerous occasions.

Another approach to evaluating routing schemes’ efficiency, is directly using them in a testbed-network, usually comprised of Raspberry Pi modules. Great examples of research incorporating hardware-based testbeds are [25–28]. Despite the existence of related work in the field of ad-hoc routing and testbed implementation, no published papers present freely available testbed frameworks. The present paper aims to fill this void by:

- Engaging in FANET-specific routing protocol and mobility model analysis

- Proposing a universal routing scheme testbed
- Making said universal testbed completely open-source for other researchers to utilize and improve

Table 1 compares the present work to already existing surveys and testbed-oriented papers. The present paper is focused not only on surveying protocols but also on proposing a testbed. Thus, only papers of a similar nature were considered for the following comparison. Our approach proves to be the most complete in covering ad hoc routing and providing an evaluation platform open for all interested researchers to use.

**Table 1.** Related work and papers.

Related Work	FANET-Specific	Routing Protocols	Mobility Models	Testbed	Open-Source
Miya et al. [25]	X	✓	X	✓	X
Kaysina et al. [26]	✓	✓	X	✓	X
Hamdaoui et al. [29]	X	✓	X	✓	X
Brown et al. [30]	✓	X	X	✓	X
Kim et al. [27]	✓	X	X	✓	X
Rosati et al. [31]	✓	✓	X	✓	X
Our approach	✓	✓	✓	✓	✓

The layout of the rest of this paper is as follows; Section 2 analyzes and explains matters of inter-UAV routing, while considering some of the most important and “influential” routing protocols. Section 3 is divided into two subsections; one surveying existing routing scheme testbeds (Section 3.1) and one proposing the aforementioned prototype testbed (Section 3.2). Section 4 is dedicated to the analysis of the proposed universal routing testbed and its modes of function: standalone, supplementary and stationary mode. Each of the aforementioned modes are explained in full detail in Sections 4.1–4.3 respectively.

## 2. Ad Hoc Routing

This section is dedicated to the analysis of the fundamental principles governing ad hoc networking, with an emphasis on FANETs. Additionally, some of the most important ad hoc routing protocols are explained and examined, in the context of evaluating them on a physical level, along with mobility models commonly used either to emulate high-mobility FANETs or as a route computation metric. The way in which mobility models affect route selection and are thus considered a route computation metric is link quality estimation (highly dependent on node mobility) accompanied by node position-awareness, which directly affects next hop selection in geographic or movement-predictive routing protocols (e.g., Greedy Perimeter Stateless Routing (GPSR) [32], Greedy Load Share Routing (GLSR) [33], Mobility Prediction based Geographic Routing (MPGR) [34], Directional Optimized Link State Routing (D-OLSR) [35], Geographic Contention-based Forwarding and Cooperative Communications (CoopGeo) [36] and their derived schemes).

### 2.1. Testbed-Friendly Ad Hoc Routing Protocols

Amongst the most commonly used ad hoc routing protocols in the scope of emulation of FANET deployments are:

- OLSR (usually implemented via OLSRd)
- Ad-hoc On-demand Distance Vector Routing (AODV)
- Hybrid Wireless Mesh Protocol (HWMP)
- BATMAN

### 2.1.1. OLSR

OLSR is one of the most popular proactive routing protocols, mostly due to the open-source nature of OLSRd enabling researchers to utilize it. Due to OLSR's popularity and configurability, it has become the foundation of a spectrum of routing protocols: Predictive OLSR (POLSR) [37], Directional OLSR (DOLSR) [35], Energy-aware Mobility Prediction OLSR (EMP-OLSR) [38], OLSR Fuzzy Cost (OLSR-FC) [39], Cross-layer OLSR (C-OLSR) [40], QoS Aware Link Defined OLSR (LD-OLSR) [41] and many more.

OLSRd can function either as classical OLSR or additionally consider the Expected Transmission Count (ETX) metric, essentially implementing OLSR-ETX to obtain link quality ( $LQ$ , network-wide metric) as well as neighbor link quality ( $NLQ$ , node-specific metric) information [29]. The ETX metric is defined as follows:

$$ETX = \frac{1}{1 - e_{pt}} \quad (1)$$

$$e_{pt} = 1 - LQ \times NLQ \quad (2)$$

$$ETX = \frac{1}{LQ \times NLQ} \quad (3)$$

where  $e_{pt}$  is the packet error probability.

When used in an actual scenario, OLSRd is responsible for detecting neighboring network nodes and performing a link quality evaluation using received signal strength indication (RSSI) information. It therefore does not handle packet transmissions, yet populates and updates nodes' routing tables as the network topology is updated by broadcasting HELLO messages and selecting MPRs. In addition to HELLO messages, another type of messages used by OLSRd is Topology Control (TC) messages, responsible for informing the network for each node's accessibility by an MPR.

Route selection in classical OLSR is implemented by considering the path consisting of the smallest number of hops. Since OLSRd can be configured to consider the ETX metric, in most cases route selection also considers the possibility of packet loss and end-to-end throughput. The ETX metric can be utilized to enable QoS awareness using OLSR [41]. The only drawback stemming from the usage of the ETX metric is the protocol's inability to sense the difference between links' data rates (it assumes that all sensed links have an identical throughput). This is counteracted by introducing the Expected Transmission Time (ETT) metric (natively implemented by OLSR-ETT) [42].

$$ETT = ETX \times \frac{S}{B} \quad (4)$$

where  $S$  = packet size in bytes and  $B$  = link capacity (measured using packet-pair technique).

### 2.1.2. AODV

AODV [43] is a reactive topology-based protocol as it calculates packet routes on-demand. It is based on the Destination-Sequenced Distance Vector routing (DSDV) [44]. AODV cannot optimize routes by default. As with OLSR, there exist -ETX variants/extensions of AODV. Researchers in [45] measured and compared the performance of AODV and AODV-ETX; they concluded that compared to AODV, AODV-ETX's average throughput is increased by 15.84%, packet delivery ratio is increased by 5.80%, average overhead is reduced by 4.31%, at the expense, however, of a 10.19% increased end-to-end delay and a by 3.09% increased route discovery delay. AODV is one of the most popular ad hoc routing protocols. Authors of [46] have compared AODV to Dynamic Source Routing (DSR) in an outdoors environment. They concluded that AODV performs measurably better in terms of packet loss and end-to-end delay. AODV is implemented in Linux kernel v. 3.8 as AODV-UU [47] created by the Uppsala University, hence the suffix.

### 2.1.3. HWMP

Using AODV as a foundation, a Hybrid Wireless Mesh Protocol (HWMP) has been developed. HWMP is a hybrid routing protocol, capable of using both proactive and reactive path selection algorithms. HWMP offers a higher packet delivery ratio, at the expense of a higher packet overhead. Similarly to AODV, HWMP uses path request (PREQ) messages to obtain optimal packet paths. Authors of [48] engage in performance comparison of AODV, DSDV, DSR, OLSR, AOMDV and HWMP alike: their measurements and simulations prove that under their test conditions, HWMP offers the best performance, followed by OLSR. HWMP's better performance is comparison to AODV stems from the aforementioned fact: usage of both proactive and reactive route path discovery methods. HWMP can route on layer 2 of the Open Systems Interconnection (OSI) stack using Media Access Control (MAC) instead of Internet Protocol (IP) addresses.

Both AODV and HWMP exist as installation for Linux-based platforms such as Raspberry pi. Due to AODV's library age, between the two, HWMP is generally preferred as a routing scheme.

### 2.1.4. BATMAN

The Better Approach To Mobile Ad hoc Networking (BATMAN) protocol is a decentralized routing protocol whose main selling point is the equal distribution of the optimal route information. This approach eliminates the need to broadcast possible topology changes to the entire network, effectively eliminating overhead. Nodes do not maintain full source-destination route information and instead only know their next hop. Route optimization is implemented in a hop-by-hop approach. Packets are routed to their destination individually, via constantly recomputed routes (that applies to packets comprising a single message as well). The BATMAN-adv variant of this protocol enables layer 2 routing (similarly to HWMP), in contrast to the classical layer-3 routing scheme normally implemented by BATMAN [49]. BATMAN-adv is currently implemented as a kernel driver and improves CPU performance by minimizing the required machine cycles [50].

Periodically, Originator Messages (OGM) are broadcasted to update neighboring node information; they contain the following information [51]:

- Originator (i.e., source) address
- Destination address
- Link flag
- Time To Live (TTL)

The information of the destination address is updated every time the OGM packet is rebroadcasted. Information regarding reachable nodes is stored in an originator table, carried by each node. BATMAN uses the Transmission Quality metric (TQ) to compare link quality between possible next hops and perform route selection [51]. Authors in [26] used BATMAN in a physical Raspberry Pi-based network. They propose a testbed framework for this protocol and additionally tested video content transmission in a BATMAN-enabled ad hoc network.

Table 2 contains and summarizes the main characteristics of the testable routing protocols which were analyzed in this subsection: OLSR Section 2.1.1, AODV Section 2.1.2, HWMP Section 2.1.3 and BATMAN Section 2.1.4.

**Table 2.** Routing scheme summary table.

Routing Protocol	Proactive	Reactive	Open-Source	Kernel Implementation
OLSR	✓	X	✓	OLSRd
AODV	X	✓	✓	AODV-UU
HWMP	✓	✓	✓	HWMP
BATMAN	✓	X	✓	BATMAN-adv

## 2.2. Mobility Models

Of the existing mobility models those supported by the NS3 simulator are mentioned in Section 1. Depending on the requirements set by the simulated UAV network use-case, as well as the behaviour defined by the routing protocol, different, and in some occasions more unique mobility models are used to describe the FANETs mobility; e.g., despite its general effectiveness, the Gauss-Markov Mobility Model does not describe circular swarm mobility (common in monitoring use-cases) as effectively as the Semi Random Circular Mobility Model, which is otherwise of minimal emulative value. Generally, the most effective (and thus popular) ones in terms of accurate FANET mobility emulation are the following:

- Gauss-Markov Mobility Model [18]
- Paparazzi Mobility Model [19]
- Semi Random Circular Mobility Model [52]
- Random Waypoint Mobility Model [18]

### 2.2.1. Gauss-Markov

Gauss-Markov is a time-dependent mobility model capable of accurately describing a FANET's movement, with individual node's paths being independent from one another. Random initial positions are defined for each UAV and the velocity vector is recomputed at each time instance depending solely on previous UAV speed and position [53].

### 2.2.2. Paparazzi

Paparazzi is a path-planned stochastic mobility model. In order for the model to approximate all possible UAV maneuvers, it supports the following fundamental movements: stay-at, way-point, eight, scan and oval [54].

### 2.2.3. Semi Random Circular

Semi Random Circular is a path-planned mobility model centered around UAV-enabled surveillance and target monitoring of an already known area [52]. This model makes the examined FANET circulate an area of interest, rendering it realistic for the emulation of e.g., search-and-rescue missions.

### 2.2.4. Random Waypoint

Random Waypoint is a randomized mobility model. At every time instance, each node has a distinct and random velocity, destination and a randomized stop time. The UAVs' destination and stop time both get re-updated upon each randomization iteration, depending on which event precedes the other: reaching of node destination or timeout of waypoint validity.

When simulating a FANET deployment, it is of utmost importance to select a fitting mobility model. As previously stated, the overall mobility of a node and its network strongly affects networking metrics and route computation-related parameters. Consideration of mobility models as a routing metric is therefore a valid reason to avoid or resort to the usage of a mobility model. Location or prediction-dependent schemes especially, may present an altered/deteriorated or generally false performance, should the UAVs' locations be altered in an unexpected/rapid manner (e.g., random definition of velocity vectors instead of a collectively coordinated mobility plan and movement path).

## 3. FANET Testbeds

This section is concerned with surveying existing routing scheme testbeds and the proposal of an open-source hardware testbed. Section 3.1 analyzes the work of researchers who faced the similar challenge of routing scheme performance evaluation in real life and under such environmental conditions. Virtually all existing work is either simulation-focused or uses Raspberry Pis on which the tested protocol is installed. The benefits of



implementing a hardware-based evaluation instead of simulating a routing scheme are numerous:

- Measurement of computational capacity required for routing and load balancing
- Measurement of environmental effects on communication and link quality
- Measurement of transmission-related node energy expenditure
- Measurement of mobility-stemming effects on overhead and computational demand
- Measurement of security methods' and packet encryption/decryption (CPU-intensive tasks) effects on availability of processing power

On a few occasions, researchers resorted to the usage of actual mobile UAVs, which they deployed as a FANET. Respectively, Section 3.2 proposes a multifunctional solution capable of both extending the capabilities of existing testbeds and functioning as a standalone on-board flight controller. The hardware design is explained on a medium abstraction layer and design files are provided.

### 3.1. Existing Routing Scheme Testbeds

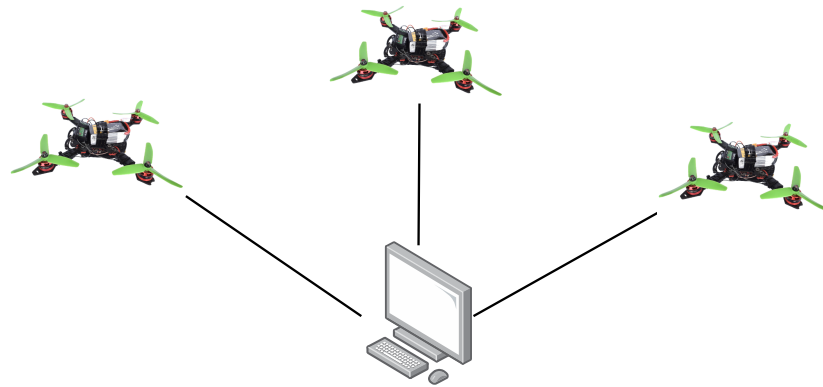
Authors of [25] performed an end-to-end packet delay measurement and evaluation of OLSR using a network comprised of stationary Raspberry Pi Zeros. Their approach proves the significant benefits of the real-life testing approach, as it enables the measurement of CPU load during routing and packet encryption. Furthermore, usage of power efficient Raspberry Pi Zeros gives insight to how a similar network would perform, should it become mobile. Additionally, usage of this platform as a routing testbed constitutes an indirect evaluation of the Linux kernel efficiency and possible bottlenecking. Random “dummy” data is generated using the *ipref* traffic generator and then embedded in a UDP frame. The source node is configured to transmit the randomly generated data to another Raspberry Pi Zero functioning as a relay between the source and destination. Due to the network being comprised of only three nodes (source, proxy, destination), no actual routing table is being made use of or updated. Therefore, usage of either a reactive/hybrid/proactive protocol has minimal effect on network metrics. Thus, this testbed implementation is more of a kernel bottlenecking evaluation rather than an actual routing protocol one. Nevertheless, repetition of this experiment using the same setup, only with a greater number of nodes would potentially wield rather important results. Similarly to this work, the researchers in [55] used five Raspberry Pis to evaluate OLSR in a content-centric network (CCN). The difference between this testbed and the one described in [25] is the usage of the networking-oriented Open-WRT operating system. Nodes maintained a constant line of sight (LOS) and proved OLSR's potential as an enabler of a CCN overlay in TCP/IP networks.

Researchers in [26] implemented a Raspberry Pi-based testbed for the BATMAN protocol. The hardware testbed was accompanied by NS-3 simulations in order for the authors to validate their results. In this case, actual UAVs were used as sources, relays and end devices alike for each tested scenario. The FANET topologies utilized were the following:

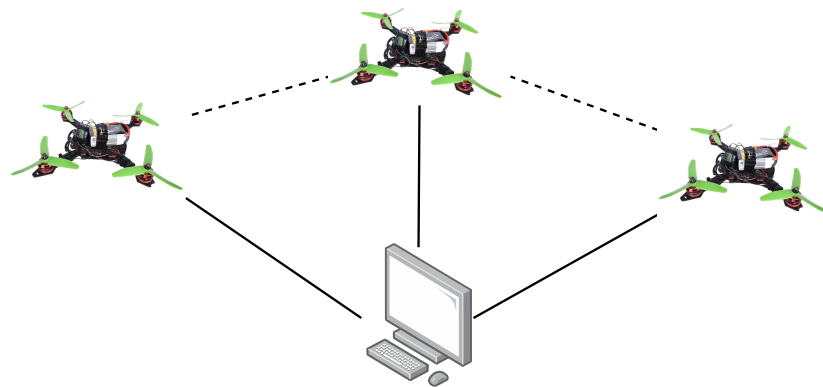
- Infrastructure topology (Figure 1)
- Mesh topology (Figure 2)
- UAV-as-a-relay topology (Figure 3)

The infrastructure topology (UAVs as end devices) (Figure 1) dictates that an intermediate ground station functions as a relay, forwarding packets to their respective destinations. Computational load of nodes is thus minimized, to the expense of the network's mobility (limited due to it being a direct function of the ground station's maneuverability). UAVs can only reach a strictly defined distance before link quality between themselves and the ground station falls below a usable threshold. The mesh topology (Figure 2) dictates that at least a UAV is constantly connected to the ground station and functions as a gateway. The difference between this topology and the previously described one is that individual UAVs are networked in a such manner that enables inter-UAV packet

relaying. This topology enables greater area coverage and a higher node mobility. The UAV-as-a-relay topology (Figure 3) is technically not a FANET topology, rather a direct UAV-enabled relaying use-case which can be extended to a FANET. It is however noteworthy due to the researchers evaluation of a such relaying method's efficiency and possible bandwidth-saving improvements.



**Figure 1.** UAVs as end devices.



**Figure 2.** UAVs in a mesh topology.



**Figure 3.** UAV as a mobile relay.

The authors of [29] used Alix.3C2 modules equipped with a Voyage (Debian-based) Linux distribution. OLSRd was installed onto the modules and the researchers evaluated both OLSR-HOPS and OLSR-ETX, with they enriched using the ETT metric as mentioned in Section 2.1.1 for higher route computation efficiency.

Researchers in [30] implemented a fixed-UAV ad hoc networking testbed and tested it using the DSR protocol. The proposed testbed is comprised of the UAVs themselves, fixed ground stations and mobile ground stations as well. The utilized hardware is comprised of the Soekris net4511 computer module accompanied by the Orinoco 802.11b Gold radio



module which has a small (PCMCIA) form factor. Fidelity Comtech RF amplifiers were used to amplify the radio module's signal which normally has an approximately 150 m range. The computer modules were equipped with the Click routing software.

### 3.2. Proposal of an Embedded Routing Scheme Testbed

There exist a variety of routing testbeds, both hardware and software in nature. However, there has been little to no progress in terms of standardization and compartmentalization of said systems. Furthermore, all hardware-based solutions seem to fall under the two following categories:

- Meant for evaluation using stationary nodes
- Specifically designed for a type of UAV the developers saw fit

This subsection is thus dedicated to the proposal of a universal and standardized hardware testbed meant for both stationary and aerial usage. Said testbed shall incorporate:

- A computing platform on which the desired routing protocol shall be installed
- A dedicated 802.11 WiFi antenna
- A gyroscope/accelerometer which shall enable the platform to function as a standalone flight controller
- A companion microcontroller implementing the flight control logic without burdening the main computing platform with unnecessary tasks

Considering the amount of existing work and availability of routing protocol installations for Linux-based operating systems, the Raspberry Pi Zero seems to be the best computing platform for aerial usage; it combines a small and lightweight form factor with impressive computing capabilities. As for the dedicated WiFi antenna and the companion microcontroller, the ESP32 proves more than sufficient in both aspects. The embedded WiFi antenna has a mediocre range, which can be significantly extended (up to approximately 1 km, assuming a direct LOS) using the Epsressif-patented 802.11 LR (long range) mode. This mode in combination with a MAC-layer peer-to-peer transmission handling protocol (e.g., ESP-NOW) can enable low latency and low overhead communication in an impressive range using a small amount of power in comparison to other commercially available 802.11 antenna modules; in this case, packet routing can be implemented using MAC addresses. The networking sacrifice needed to be made is throughput. Entering the LR mode is a simple function call and does not affect program functionality:

```
esp_wifi_set_protocol(ifx , WIFI_PROTOCOL_LR);
```

Table 3 showcases the specifications of both the computer and complementary microcontroller of choice. The gyroscope/accelerometer module of choice is the MPU6050 due to the amount and quality of existing libraries enabling it to interface with ESP32. More technical details and design choices are thoroughly explained below.

**Table 3.** Raspberry Pi Zero W & ESP32 specifications.

	Raspberry Pi Zero W	ESP32
CPU	ARM1176JZF-S (1 GHz, single-core)	Xtensa LX6 (240 MHz, dual-core)
Memory	RAM 512 MB (LPDDR2)	520 KB (SRAM) 4 MB (flash)

The proposed testbed architecture makes use of affordable networking and computing equipment to create a mobile ad hoc network and evaluate routing efficiency of a given scheme. In addition to route computation and transmission handling, this implementation can be used as a standalone flight controller. The embedded gyroscope/accelerometer enables each module to not only function as a self-contained flight controller but also consider orientation and angular velocity of each axis as a routing metric; this enables

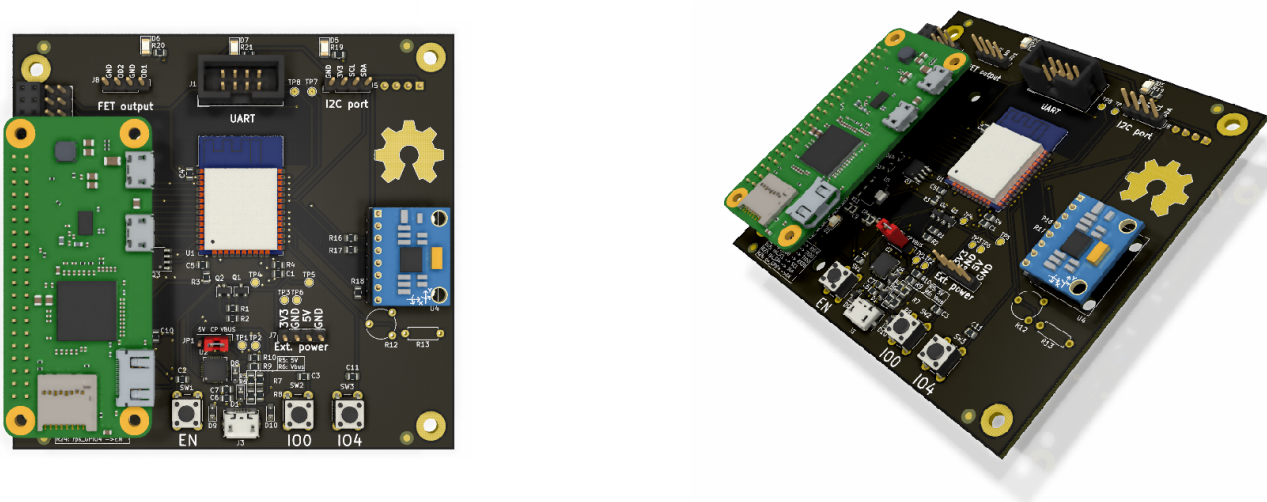
support for location-predictive routing protocols. Since the module is centered around and uses the Raspberry Pi as a computing unit, all the respectively supported routing protocols may be used with this design. Each module is a functional node in itself and does not require additional components.

The hardware has been designed so that each one of said testbed modules can be directly attached to an existing UAV. The main board complies to the PC/104 standard, which is the hardware specification many UAV autopilots aim to conform to as well, as mentioned in [56]. Conformity to this electromechanical standard ensures flawless mechanical interfacing and even safe physical stacking of said modules onto the same UAV. PC/104 is commonly used in specialized high-reliability embedded applications, aiming for a small and rugged form-factor. The PC/104 standard dictates the following:

- Usage of a (traditionally 104-pin) connector
- Strictly defined PCB dimensions ( $90 \times 96$  mm) with 13 mm optional extra space for additional connectors
- Four mounting holes in predefined positions

#### 4. Findings and Discussion

The Findings and Discussion section is dedicated to the explanation of the three main modes in which the proposed module shall operate. The first mode is a result of direct usage of the proposed module as both a flight controller and a routing evaluation medium. As for the second mode, it likely constitutes the most directly usable mode of operation: the module only implements routing (and the evaluation thereof), while an external flight controller, native to the UAV, handles all aircraft control, PID tuning and external sensor interfacing. Similarly, the last mode of operation is the one in which no aircraft is utilized for scheme evaluation, rather a stationary set of networked modules. The last mode is suitable for both VANET deployments as well as stationary (e.g., sensor network) networks. Figure 4 showcases a 3D render of the design.



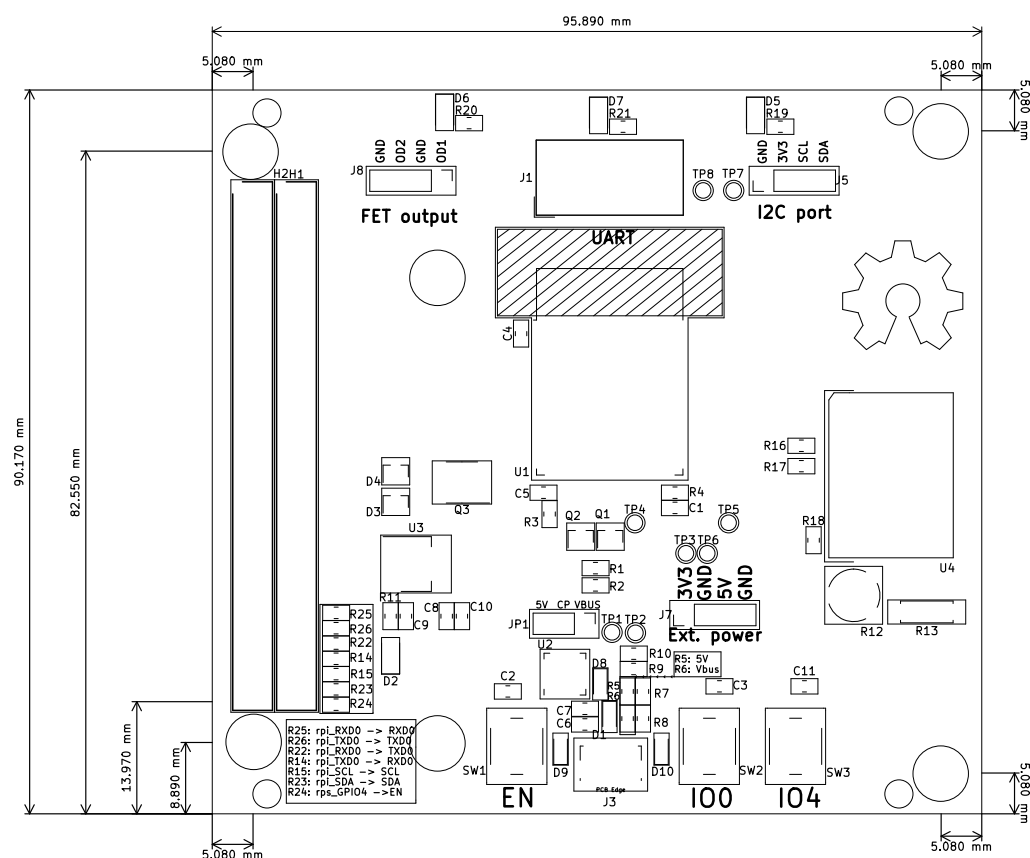
**Figure 4.** Rendering of proposed testbed module.

The design files for the proposed testbed module have been made publicly available in GitHub: [57]. The design software used is KiCAD. Along with the design files are the production file, which can be used for fast prototyping and testing. The work of engineers in [58] was utilized as a PC/104 template.

##### 4.1. Standalone Aerial Mode

In standalone aerial mode, mesh modules are directly attached to each UAV comprising a FANET. The main concept behind the design is the compartmentalization of task

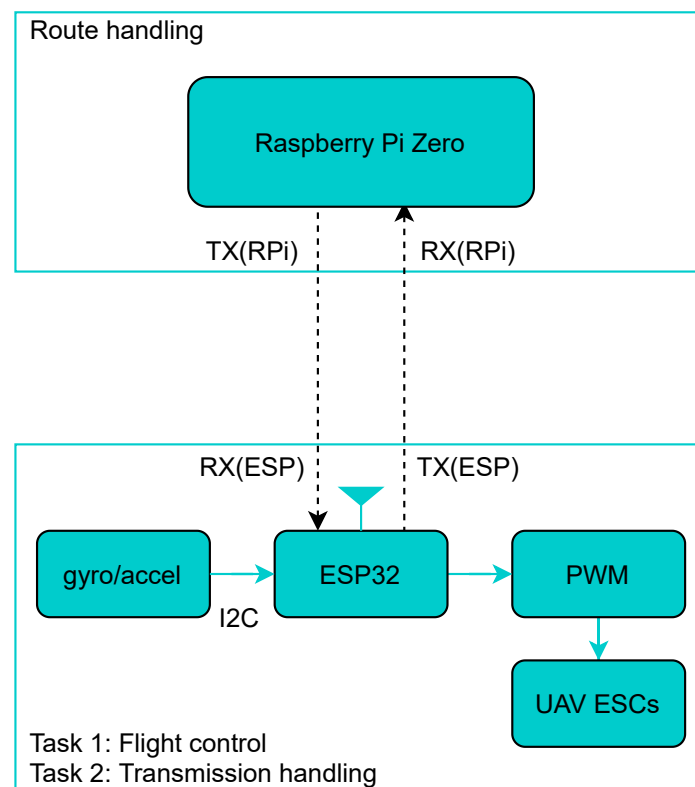
handling so as to avoid unnecessary computational burden of each component. This design idea is directly translated to hardware design implementation. The Raspberry Pi Zero is attached directly to the main board with an exposed header configured to match its pinout. As seen in Figure 5, one of the PC/104 holes and two extra nonplated 3.2 mm  $\varnothing$  mounting holes offer extra mechanical stability to the interconnection of the main components. As the Raspberry Pi is placed onto the main board, it is internally powered by the main board's regulator. The module has full support for an amplitude of commercially available GNSS modules. Node location can therefore be used as a route selection metric, in addition to node velocity and acceleration as previously mentioned. Thus, in addition to predictive routing schemes, evaluation/implementation of location aware or location-based routing is made possible.



**Figure 5.** Layout of the module's main board.

Figure 6 illustrates the interfacing between the Raspberry Pi and the ESP32 as well as tasks each module handles.

The Raspberry Pi Zero shall only implement route-related computations. After a route has been established, the packet shall be serially transmitted to the ESP32, which shall in turn implement transmission handling. In turn, upon packet reception, the ESP32 shall serially transmit the data to the Raspberry Pi. Packet transmission/reception handling shall constitute a distinct nonblocking RTOS task, so that the ESP32 can simultaneously implement the flight control logic. It becomes obvious that the ESP32 now has two functions, which are defined as distinct RTOS tasks handled by the two cores of the Xtensa LX6 microprocessor. All packet transmission and reception is done via the embedded WiFi antenna which ensures a low power consumption as well as the possibility of interfacing with conventional 802.11 networks. The Raspberry Pi is thus successfully only occupied with route handling.



**Figure 6.** Standalone aerial node components.

Table 4 showcases the serial interface between said modules in standalone aerial mode. The mentioned gyroscope/accelerometer processed data shall only be transmitted to the Raspberry Pi if the evaluated routing protocol requires them as a metric. Similarly, RSSI (measured between) shall be shared if link quality or power-awareness is desired and considered by the respective protocol. GNSS data is again, optional and shall be transmitted in the case geolocation-based routing is implemented.

**Table 4.** Raspberry Pi Zero W—ESP serial interfacing.

Raspberry Pi Zero W		ESP32	
Serial Tx	Serial Rx	Serial Tx	Serial Rx
Packet to be transmitted	Received packet	Received packet	Packet to be transmitted
	Gyroscope/accelerometer (processed data)	Gyroscope/accelerometer (processed data)	
	RSSI	RSSI	
	GNSS data	GNSS data	

#### 4.2. Supplementary Aerial Mode

In supplementary aerial mode, the module does not function as a flight controller. It may be attached to an already existing flight controller/autopilot and solely implement routing to other networked nodes on-board. The authors of [56] survey a number of UAV autopilots, a substantial number of whom offer serial interface connectivity. Flight controllers typically transmit telemetry data to a ground station, which can be made available to a serially connected interface on demand. Telemetry data can be forwarded to the Raspberry Pi to be used in route calculations. In this case, the ESP does not need to

handle two distinct RTOS tasks and can fully commit to the task of transmission handling. The interfacing node components can be described as seen in Figure 7.

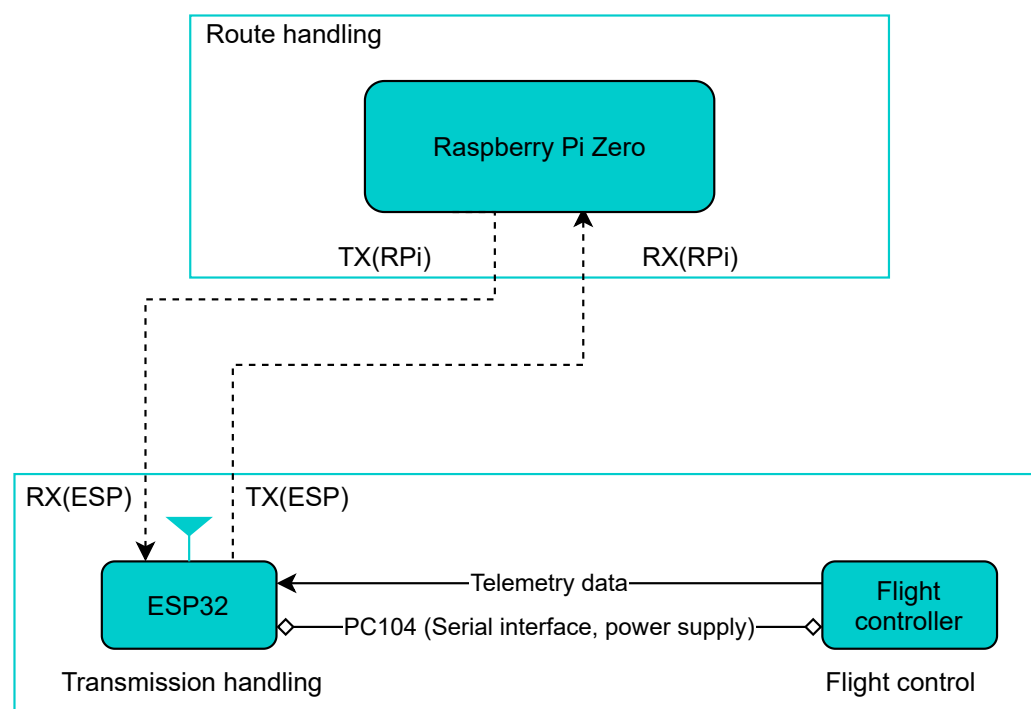


Figure 7. Supplementary aerial mode components.

As virtually all flight controller inherently implement interfacing with GNSS modules, and contain an on-board gyroscope/accelerometer module, those components need not be placed/connected to the main board.

#### 4.3. Stationary/VANET Mode

In stationary/VANET mode, the data from the MPU6050 gyroscope/accelerometer module may be utilized to form packets of a given length, defined by the protocol's structure. In this mode, the modules shall function independently from an external flight controller. With no connection to a flight controller or an embedded flight control RTOS task, the module can still implement routing and be used as a means of evaluation thereof. This may prove to be a useful implementation for non MANET-specific ad hoc routing schemes. The advantages stemming from the usage of a compartmentalized device and a standard form factor are still there, though exploited to a lesser degree. Usage of this mode is also a possibility for the evaluation of routing schemes in nonaerial ad hoc networks.

### 5. Conclusions and Future Work

This paper has discussed industrial, military and agricultural use-cases for FANETs, surveyed directly applicable mobile ad hoc routing protocol to be used in such deployments, as well as existing testbed/simulation frameworks which can be used to evaluate the efficiency of routing protocols. Additionally, matters of network topology and mobility modeling are addressed in the context of scheme simulation. This work is concluded with the presentation of:

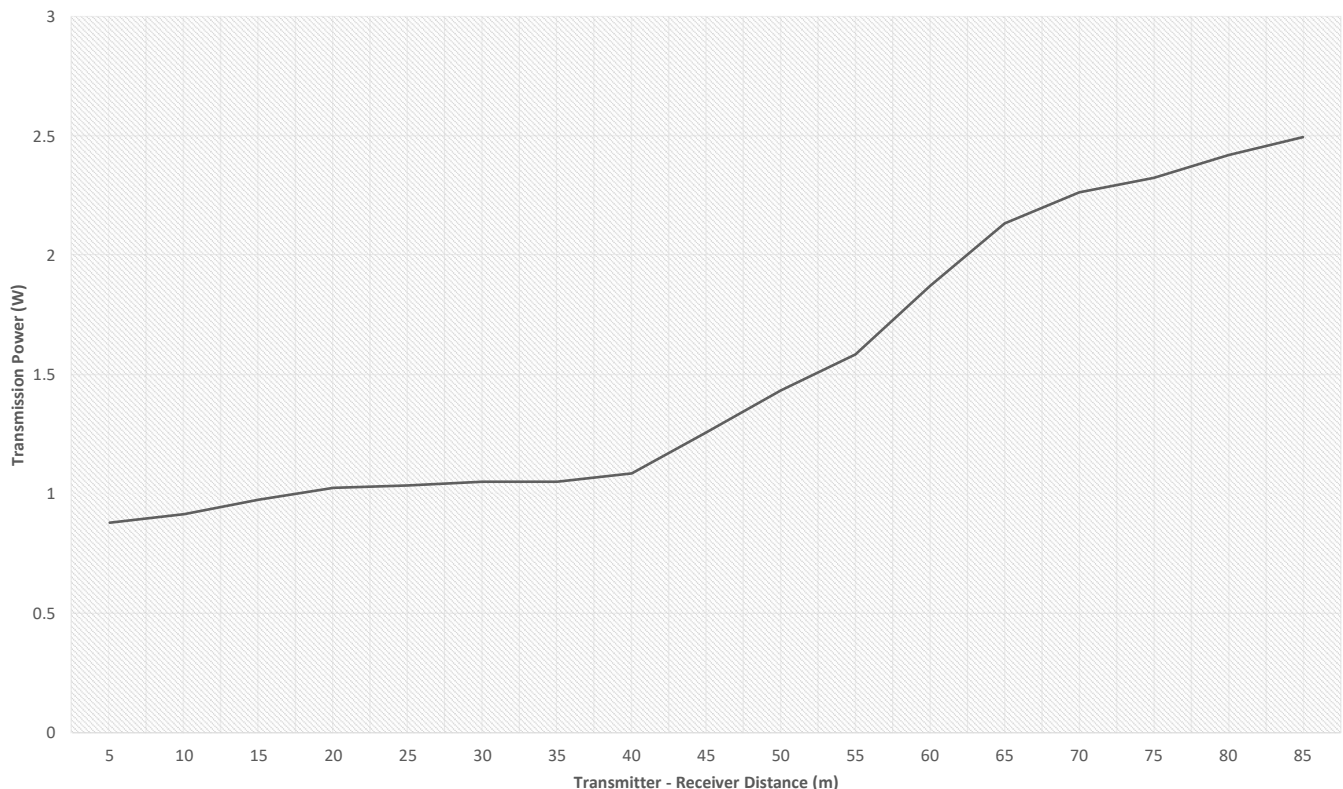
- A standardized hardware prototype capable of supporting all routing protocols currently implemented in the Linux kernel
- Component interface modes as a function of desired evaluation framework

Figure 8 constitutes the result of the proposed testbed module's evaluation using an ESP-Now link and the embedded PCB antenna of the ESP. Even when using the PCB



antenna, power consumption was kept below 2.5 Watts, while the link did not fail to maintain maximum bandwidth at 85 m with a direct and constant LOS. The module's evaluation consisted of:

- Link establishment between two modules
- Formation of ESP-Now packets using dummy data harvested from the MPU6050 gyroscope/accelerometer
- Communication between a constantly increasing distance (5 m–85 m)
- Measurement of the transmitting module's power consumption at 5-m intervals



**Figure 8.** Communication power of an ESP-Now direct link between two modules.

Future work shall be constituted of test results originating from the usage of the proposed testbed with OLSRd and BATMAN. A comparison between said routing schemes in terms of: (a) scalability and (b) mobility shall be part of said future work. NS-3 simulations shall be used to validate the performance of the proposed testbed under the mobility models mentioned in Section 2.2. Due to the current lack of functional open-source flight controllers meant to be used with the ESP-series, such firmware shall be developed and made openly available, as is the case with the proposed module's hardware.

**Author Contributions:** Conceptualization, T.L., P.S., V.V., P.F. and G.A.; methodology, G.A.; software, G.A.; validation, T.L., G.A. and P.S.; formal analysis, G.A., T.L., P.S.; investigation, G.A.; resources, T.L., P.S., V.V.; data curation, T.L., P.F., V.V.; writing—original draft preparation, G.A.; writing—review and editing, T.L., G.A.; visualization, G.A.; supervision, T.L. and P.S.; project administration, T.L. and P.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was cofunded by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, grant number T1EDK-04759.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.



**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Oubbati, O.S.; Atiquzzaman, M.; Lorenz, P.; Tareque, M.H.; Hossain, M.S. Routing in flying Ad Hoc networks: Survey, constraints, and future challenge perspectives. *IEEE Access* **2019**, *7*, 81057–81105. [\[CrossRef\]](#)
- Khan, M.F.; Yau, K.L.A.; Noor, R.M.; Imran, M.A. Routing schemes in FANETs: A survey. *Sensors* **2020**, *20*, 38. [\[CrossRef\]](#)
- Nawaz, H.; Ali, H.M.; Laghari, A.A. UAV Communication Networks Issues: A Review. *Arch. Comput. Methods Eng.* **2020**, 1–21. [\[CrossRef\]](#)
- Di Felice, M.; Trotta, A.; Bedogni, L.; Chowdhury, K.R.; Bononi, L. Self-organizing aerial mesh networks for emergency communication. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, Washington, DC, USA, 2–5 September 2014; pp. 1631–1636. [\[CrossRef\]](#)
- Maza, I.; Caballero, F.; Capitán, J.; Martínez-De-Dios, J.R.; Ollero, A. Experimental results in multi-UAV coordination for disaster management and civil security applications. *J. Intell. Robot. Syst. Theory Appl.* **2011**, *61*, 563–585. [\[CrossRef\]](#)
- Nikitas, A.; Michalakopoulou, K.; Njoya, E.T.; Karampatzakis, D. Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era. *Sustainability* **2020**, *12*, 2789. [\[CrossRef\]](#)
- Cameron, S.; Hailes, S.; Julier, S.; McClean, S.; Parr, G.; Trigoni, N.; Ahmed, M.; McPhillips, G.; de Nardi, R.; Nie, J.; et al. SUAAVE: Combining aerial robots and wireless networking. In Proceedings of the 25th Bristol International UAV Systems Conference, Department of Aerospace Engineering, University of Bristol, 1–9 Old Park Hill, Bristol, Avon, UK, 12–14 April 2010; pp. 1–14.
- Ollero, A.; Kondak, K.; Previnaire, E.; Maza, I.; Caballero, F.; Bernard, M.; Martinez, J.R.; Marrón, P.; Herrmann, K.; Van Hoesel, L.; et al. Integration of aerial robots and wireless sensor and actuator networks. The AWARE project. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 1104–1105. [\[CrossRef\]](#)
- Lagkas, T.; Argyriou, V.; Bibi, S.; Sarigiannidis, P. UAV IoT framework views and challenges: Towards protecting drones as “things”. *Sensors* **2018**, *18*, 4015. [\[CrossRef\]](#) [\[PubMed\]](#)
- Radoglou-Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios, I. A compilation of UAV applications for precision agriculture. *Comput. Netw.* **2020**, *172*, 107148. [\[CrossRef\]](#)
- Kakamoukas, G.; Sarigiannidis, P.; Livanos, G.; Zervakis, M.; Ramnalis, D.; Polychronos, V.; Karamitsou, T.; Folinas, A.; Tsitsiokas, N. A Multi-collective, IoT-enabled, Adaptive Smart Farming Architecture. In Proceedings of the IST 2019—IEEE International Conference on Imaging Systems and Techniques, Abu Dhabi, UAE, 9–10 December 2019. [\[CrossRef\]](#)
- Triantafyllou, A.; Sarigiannidis, P.; Bibi, S. Precision agriculture: A remote sensing monitoring system architecture. *Information* **2019**, *10*, 348. [\[CrossRef\]](#)
- Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [\[CrossRef\]](#)
- Egarguin, N.J.; Jackson, D.R.; Onofrei, D.; Leclerc, J.; Becker, A. Adaptive beamforming using scattering from a drone swarm. In Proceedings of the 2020 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems: Making Waves in Texas, WMCS 2020, Waco, TX, USA, 26–28 May 2020. [\[CrossRef\]](#)
- Mann, S.; Pierce, C.; Hernandez, J.; Li, Q.; Zheng, B.C.; Xin Xiang, Y. Drone Swarms for Sensing-of-Sensing. *Proc. IEEE Sens.* **2019**, *2019*, 23–26. [\[CrossRef\]](#)
- Sahingoz, O.K. Networking models in flying Ad-hoc networks (FANETs): Concepts and challenges. *J. Intell. Robot. Syst. Theory Appl.* **2014**, *74*, 513–527. [\[CrossRef\]](#)
- Guo, Y.; Li, X.; Yousefi-zadeh, H.; Jafarkhani, H. UAV-aided cross-layer routing for MANETs. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC, Paris, France, 1–4 April 2012; pp. 2928–2933. [\[CrossRef\]](#)
- Kuiper, E.; Nadjm-Tehrani, S. Mobility models for UAV group reconnaissance applications. In Proceedings of the 2006 International Conference on Wireless and Mobile Communications (ICWMC'06), Bucharest, Romania, 29–31 July 2006; pp. 2–8. [\[CrossRef\]](#)
- Camp, T.; Boleng, J.; Davies, V. A survey of mobility models for ad hoc network research. *Wirel. Commun. Mob. Comput.* **2002**, *2*, 483–502. [\[CrossRef\]](#)
- Guillen-Perez, A.; Cano, M.D. Flying ad hoc networks: A new domain for network communications. *Sensors* **2018**, *18*, 3571. [\[CrossRef\]](#) [\[PubMed\]](#)
- Broch, J.; Maltz, D.A.; Johnson, D.B.; Hu, Y.C.; Jetcheva, J. A performance comparison of multi-hop wireless ad hoc network routing protocols, In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking New York, NY, USA, 25 October 2001; pp. 85–97. [\[CrossRef\]](#)
- Reed, J.M.; Abdallah, A.S.; Thompson, M.S.; MacKenzie, A.B.; DaSilva, L.A. The FINS Framework: Design and Implementation of the Flexible Internetwork Stack (FINS) Framework. *IEEE Trans. Mob. Comput.* **2016**, *15*, 489–502. [\[CrossRef\]](#)
- Thompson, M.; Abdallah, A.; Reed, J.; MacKenzie, A.; Dasilva, L. The FINS framework: An open source userspace networking subsystem for linux. *IEEE Netw.* **2014**, *28*, 32–37. [\[CrossRef\]](#)
- Abdallah, A.S.; Horvath, M.D.; Thompson, M.S.; MacKenzie, A.B.; Dasilva, L.A. Facilitating experimental networking research with the FINS framework. In Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, New York, NY, USA, 19 September 2011; pp. 103–104. [\[CrossRef\]](#)

25. Miya, T.; Ohshima, K.; Kitaguchi, Y.; Yamaoka, K. Experimental Analysis of Communication Relaying Delay in Low-Energy Ad-hoc Networks *arXiv* **2020**, arXiv:2010.15572.
26. Kaysina, I.A.; Vasiliev, D.S.; Abilov, A.; Meitis, D.S.; Kaysin, A.E. Performance evaluation testbed for emerging relaying and coding algorithms in Flying Ad Hoc Networks. In Proceedings of the Moscow Workshop on Electronic and Networking Technologies, MWENT, Moscow, Russia, 14–16 March 2018; pp. 1–5. [\[CrossRef\]](#)
27. Kim, G.H.; Nam, J.C.; Mahmud, I.; Cho, Y.Z. Multi-drone control and network self-recovery for flying Ad Hoc Networks. In Proceedings of the International Conference on Ubiquitous and Future Networks, ICUFN, Vienna, Austria, 5–8 July 2016; pp. 148–150. [\[CrossRef\]](#)
28. Shan, F.; Luo, J.; Xiong, R.; Wu, W.; Li, J. Looking before Crossing: An Optimal Algorithm to Minimize UAV Energy by Speed Scheduling with a Practical Flight Energy Model. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 6–9 July 2020; pp. 1758–1767. [\[CrossRef\]](#)
29. Hamdaoui, B.; Sinky, H. Optimized link state routing for quality-of-service provisioning: Implementation, measurement, and performance evaluation. *Wirel. Commun. Mob. Comput.* **2015**, *14*, 421–430. [\[CrossRef\]](#)
30. Brown, T.; Doshi, S.; Jadhav, S.; Henkel, D.; Thekkekkunnel, R. A full scale wireless ad hoc network test bed. *Access* **2005**, *7*, 51–60.
31. Rosati, S.; Kruzelecki, K.; Heitz, G.; Floreano, D.; Rimoldi, B. Dynamic Routing for Flying Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 1690–1700. [\[CrossRef\]](#)
32. Karp, B.; Kung, H.T. Greedy Perimeter Stateless Routing for Wireless Networks(GPSR ). In Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, Boston, MA, USA, 1 August 2000. [\[CrossRef\]](#)
33. Medina, D.; Hoffmann, F.; Rossetto, F.; Rokitansky, C.H. North atlantic inflight internet connectivity via airborne mesh networking. In Proceedings of the IEEE Vehicular Technology Conference, San Francisco, CA, USA, 5–8 September 2011. [\[CrossRef\]](#)
34. Lin, L.; Sun, Q.; Wang, S.; Yang, F. A geographic mobility prediction routing protocol for Ad Hoc UAV Network. In Proceedings of the 2012 IEEE Globecom Workshops, GC Wkshps 2012, Anaheim, CA, USA, 3–7 December 2012; pp. 1597–1602. [\[CrossRef\]](#)
35. Alshabtat, A.I.; Dong, L. Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks. *World Acad. Sci. Eng. Technol.* **2011**, *80*, 705–711.
36. Aguilar, T.; Ghedira, M.C.; Syue, S.J.; Gauthier, V.; Afifi, H.; Wang, C.L. A Cross-Layer Design Based on Geographic Information for Cooperative Wireless Networks. In Proceedings of the 2010 IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, 16–19 May 2010; pp. 1–5. [\[CrossRef\]](#)
37. Rosati, S.; Kruzelecki, K.; Traynard, L.; Rimoldi, B. Speed-aware routing for UAV ad-hoc networks. In Proceedings of the 2013 IEEE Globecom Workshops, GC Wkshps 2013, Atlanta, GA, USA, 9–13 December 2013; pp. 1367–1373. [\[CrossRef\]](#)
38. Liu, J.; Ji, X.; Huang, C.; Tan, X. A novel OLSR protocol with mobility prediction. In Proceedings of the Third International Conference on Cyberspace Technology (CCT 2015), Beijing, China, 17–18 October 2015. [\[CrossRef\]](#)
39. José, D.A.M.; Bulcão-Neto, R.F.; Patto, V.S.; Sene Júnior, I.G. OLSR fuzzy cost (OLSR-FC): An extension to olsr protocol based on fuzzy logic and applied to avoid selfish nodes. *Rev. Inform. Teórica Apl.* **2019**, *26*, 60–77. [\[CrossRef\]](#)
40. Fan, X.; Lin, J.; Zhang, C.; Zhang, S. A cross-layer anti-jamming routing protocol for FANETS. In Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications, ICC3 2018, Chengdu, China, 7–10 December 2018; pp. 301–305. [\[CrossRef\]](#)
41. Jain, R.; Kashyap, I. An QoS Aware Link Defined OLSR (LD-OLSR) Routing Protocol for MANETS. *Wirel. Pers. Commun.* **2019**, *108*, 1745–1758. [\[CrossRef\]](#)
42. Lavén, A.; Hjältquist, P. *Multimetric OLSR and ETT Case Study Final Report Multimetric OLSR and ETT 1 Introduction*; Karlstad University: Karlstad, Sweden, 2014.
43. Perkins, C.E.; Royer, E.M. Ad-hoc on-demand distance vector routing. In Proceedings of the WMCSA'99: 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, USA, 25–26 February 1999; pp. 90–100. [\[CrossRef\]](#)
44. Perkins, C.E.; Watson, T.; Bhagwat, P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *ACM SIGCOMM Comput. Commun. Rev.* **1994**, *24*, 234–244. [\[CrossRef\]](#)
45. Purnomo, A.; Widyawan; Najib, W.; Hartono, R.; Hartatik. Performance Comparison of Modified AODV-ETX with AODV and AODV-ETX Routing Protocol in an MANET. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *578*, 012082. [\[CrossRef\]](#)
46. Hou, S.; Wu, M.; Liao, W.; Wang, D. Performance comparison of AODV and DSR in MANET test-bed based on Internet of Things. In Proceedings of the 2015 IEEE 82nd Vehicular Technology Conference, VTC Fall 2015, Boston, MA, USA, 6–9 September 2015; pp. 3–7. [\[CrossRef\]](#)
47. Jung, S.; Kim, B.; Kim, K.; Roh, B.; Ham, J. Implementation of AODV-UU on Linux 4.15 Kernel. In Proceedings of the 2019 IEEE 16th International Conference on Mobile Ad Hoc and Smart Systems Workshops, MASSW 2019, Monterey, CA, USA, 4–7 November 2019; pp. 160–161. [\[CrossRef\]](#)
48. Nayyar, A. Flying Adhoc Network (FANETS): Simulation Based Performance Comparison of Routing Protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP. In Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems, icABCD 2018, Durban, South Africa, 6–7 August 2018; pp. 1–9. [\[CrossRef\]](#)
49. Chissungu, E.; Blake, E.; Le, H. Performance Comparison of B.A.T.M.A.N.d and B.A.T.M.A.N.-adv. In Proceedings of the Southern Africa Telecommunication Networks and Applications Conference, East London, South Africa, 3–7 September 2011; pp. 6–7.
50. Lagkas, T.; Imeri, A.; Eleftherakis, G. *A Pragmatic Evaluation of Distance Vector Proactive Routing in MANETs via Open Space Real-World Experiments*; Springer: Cham, Switzerland, 2015.

- 
51. Chissungu, E.; Blake, E.; Le, H. Investigation into Batmand-0.3.2 protocol performance in an indoor Mesh Potato testbed. In Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012, Fukuoka, Japan, 26–29 March 2012; pp. 526–532. [[CrossRef](#)]
  52. Wang, W.; Guan, X.; Wang, B.; Wang, Y. A novel mobility model based on semi-random circular movement in mobile ad hoc networks. *Inf. Sci.* **2010**, *180*, 399–413. [[CrossRef](#)]
  53. Meghanathan, N. Impact of the Gauss-Markov mobility model on network connectivity, lifetime and hop count of routes for mobile Ad hoc networks. *J. Netw.* **2010**, *5*, 509–516. [[CrossRef](#)]
  54. Bouachir, O.; Abrassart, A.; Garcia, F.; Larrieu, N. A mobility model for UAV ad hoc network. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014, Orlando, FL, USA, 27–30 May 2014; pp. 383–388. [[CrossRef](#)]
  55. Barolli, A.; Oda, T.; Barolli, L.; Takizawa, M. Experimental results of a raspberry Pi and OLSR based wireless content centric network testbed considering OpenWRT OS. In Proceedings of the International Conference on Advanced Information Networking and Applications, AINA, Crans-Montana, Switzerland, 23–25 March 2016; pp. 95–100. [[CrossRef](#)]
  56. Chao, H.; Cao, Y.; Chen, Y. Autopilots for Small Fixed-Wing Unmanned Air Vehicles: A Survey. In Proceedings of the 2007 International Conference on Mechatronics and Automation, Harbin, China, 5–8 August 2007; Technical Report.
  57. Amponis, G. Mesh Module GitHub Repository. 2020. Available online: [https://github.com/g-ampo/mesh\\_module](https://github.com/g-ampo/mesh_module) (accessed on 27 December 2020).
  58. LibreCube. LibreCube PC/104 Template. 2019. Available online: <https://gitlab.com/librecube/support/templates/librecube-board> (accessed on 27 December 2020).