# IEEE 802.11n vs. IEEE 802.15.4: a study on Communication QoS to provide Safe FANETs

Emerson A Marconato, Daniel F Pigatto, Kalinka R L J Castelo Branco, Jean-Aimé Maxa, Nicolas Larrieu, Alex S R Pinto

# IEEE 802.11n vs. IEEE 802.15.4: a study on Communication QoS to provide Safe FANETs

Emerson A. Marconato,
Daniel F. Pigatto and
Kalinka R. L. J. Castelo Branco
Institute of Mathematics and
Computer Sciences,
University of São Paulo,
São Carlos, São Paulo, Brazil
Emails: {emerson,pigatto,
kalinka}@icmc.usp.br

Jean Aimé Maxa,
Nicolas Larrieu
Ecole Nationale de l'Aviation Civile,
Toulouse, France
Emails: maxa@recherche.enac.fr
nicolas.larrieu@enac.fr

Alex S. R. Pinto
Federal University of Santa Catarina
Blumenau, Santa Catarina
Email: a.r.pinto@ufsc.br

***Abstract*—Flying Ad hoc Network (FANET) is an infrastructure-less multi-hop radio ad hoc network in which Unmanned Aerial Vehicles (UAVs) and Ground Control Station (GCS) collaborates to forward data traffic. Compared to the standard Mobile Ad hoc NETworks (MANETs), the FANET architecture has some specific features (3D mobility, low UAV density, intermittent network connectivity) that bring challenges to the communication protocol design. Such routing protocol must provide safety by finding an accurate and reliable route between UAVs. This safety can be obtained through the use of agile method during software based routing protocol development (for instance the use of Model Driven Development) by mapping each FANET safety requirement into the routing design process. This process must be completed with a sequential safety validation testing with formal verification tools, standardized simulator (by using real simulation environment) and real-world experiments. In this paper, we considered FANET communication safety by presenting design methodologies and evaluations of FANET routing protocols. We use the LARISSA architecture to guarantee the efficiency and accuracy of the whole system. We also use the model driven development methodology to provide model and code consistency through the use of formal verification tools. To complete the FANET safety validation, OMNeT++ simulations (using real UAVs mobility traces) and real FANET outdoor experiments have been carried out. We confront both results to evaluate routing protocol performances and conclude about its safety consideration.**

***Keywords*—Dynamic routing safety, Flying ad hoc networks, Unmanned Aerial Vehicle, safety validation**

## I. Introduction

Advances in Unmanned Aerial Vehicle (UAV) technologies are allowing Flying Ad Hoc Networks (FANETs) a reality. However in order to achieve an effective cooperation among multiple UAVs, it is necessary to model distinct communication protocols. Basically, a FANET can be considered a robot ad-hoc network (when no infrastructure is used) or a robot sensor network (when a ground station is considered).

The main idea of FANET is to perform cooperative sensing, using multiple UAVs to cover an area that is not possible to be covered by a single UAV. Thus, it is necessary to have a reliable communication and to maintain QoS issues. There are several research efforts in robot sensor networks and several challenges are faced, such as: robot control, robot localization and communication QoS. Therefore, these challenges are very similar in FANETs.

FANETs can be used in several applications e. g. agriculture (multiple UAVs can cooperate to monitor or actuate in precision agriculture); goods delivering; and defense (FANETs can be used in coordinated critical missions).

In this paper, we will consider a FANET infrastructure composed of multiple UAVs and one ground station. Different simulations and prototype tests are presented and discussed. The main scientific question that guides this paper is: which FANET topology is more QoS communication effective? Thus, we tested and simulated FANET scenarios based on a **star** network topology (all UAVs directly communicating with the ground station) and on **mesh** topology (a dynamic routing is necessary). The simulations were based on real scenarios traces where parameters like speed and mobility pattern were extracted from real-world experiments.

Furthermore, the large amount of UAVs (that currently are not certified) must be inserted into a non-segregated airspace. Therefore, a method that certifies the development of these unmanned aircraft (in hardware and software level) must be used. Moreover, the communication of UAV-UAV, UAV-GCS (Ground Control Station) must be certified too. Therefore, LARISSA guarantees this certification achievement, so it was used in the concept and simulations of this paper aiming at achieving such certification.

LARISSA is a layered architecture model that interconnects unmanned systems [1]. This architecture model splits components of an Unmanned Aircraft System into aerial and ground segments. The aerial segment is hierarchically composed of six layers: i) physical, ii) distributed RTOS (Real Time Operating System), iii) system abstraction, iv) monitoring and control, v) navigation & services, and vi) mission layer. The ground segment is divided into the i) physical layer and ii) ground control station layer. These layers can be represented by models that guide the development, specifying how to interconnect the various components such as sensors, control circuits, GPS (Global Position System),

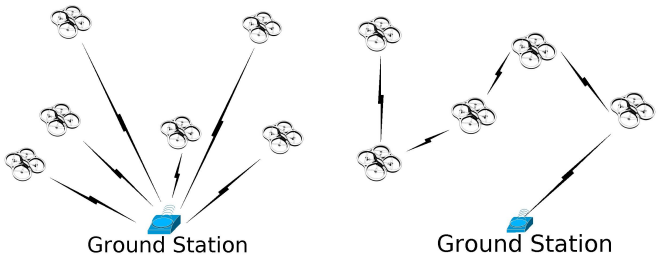
payload, communication with the ground control station, and others.

### A. Problem Statement

Communication is a crucial element for safety, which is considered part of dependability [2]. The main reason is that UAVs must maintain messages exchange in order to coordinate a mission. Therefore, it is necessary to define a suitable UAV network topology that achieves the QoS level necessary to keep connectivity. Thus, we consider a FANET composed of N UAVs and 1 ground station. The ground station must receive a certain amount of control packets of all UAVs during a period.

This way, the main question we want to answer is ***what kind of FANET topology is the most efficient, mesh or star?***

In a star FANET (Fig. 1a) all UAVs will send their messages to the ground station in one-hop. On the other hand, in a mesh FANET (that must have a dynamic routing protocol - Fig. 1b) UAVs will route messages until the ground station is reached.




(a) Star FANET. (b) Mesh FANET.

Fig. 1: FANET topology.

Recently has been observed that UAVs are increasingly working in cooperation, flying in formation or simply collaborating, which in fact necessarily introduces the need for communication among them and also with ground stations and/or satellites.

Routing algorithms play an important role in connectivity since broadcasting messages can generate unnecessary traffic on the network and traffic congestion. Thus, the application routing algorithms becomes a need to ensure connectivity and hence increasing the safety of the UAV and all the elements that compose an unmanned aircraft system, emphasizing that mobility pattern of FANETs has a strong influence on communication between UAVs.

Finally, this study indicates which is the most suitable topology (mesh or star) in order to guarantee safe FANET communication based in real experiments and simulation scenarios.

The main contributions of this paper are: (i) a comparison of **star** and **mesh** FANET showing which is safer; (ii) an automatic OMNeT++ code generation using a model architecture (LARISSA), which guarantees the real modeling and standards information about the aircraft, once safe unmanned avionic operations require a detailed analysis of the communication aspects; (iii) a methodology using Model Driven Development (MDD) to model and test routing algorithms; (iv) real experiments prototypes compared with simulation results.

## II. Related Works

A challenging issue in cooperation with multiple unmanned aerial vehicles (UAVs) deployment is the efficient networking of the UAVs over the wireless medium in quickly changing environments. Typically, a FANET communication architecture is defined by a set of mechanisms that determines how data traffic flows between the GCS and the multiple UAVs. To the best of our knowledge, there is no previous work that addresses FANET routing protocols while making bridge to its safety validation. However, some papers that are considered to be among the first works in the field have specially focused on FANET communication architecture design challenges and issues. Among these papers, we quote [3], in which Bekmezci et al. were interested in the concept and the challenge of creating a FANET architecture.

The authors in [4] present a summary of four main types of communication architectures: i) *Centralized UAV Network* (it has a central node i.e., the ground station, to which all UAVs are connected); ii) *UAV ad hoc network* (it does not rely on a pre-existing infrastructure, and each UAV will participate in data forwarding for other UAVs of the network); iii) *Multi-Group UAV Network* (UAVs within a group form a UAV ad hoc network with its respective backbone UAV connecting to the ground station; iv) *Multi-Layer UAV Ad Hoc Network* (the UAVs within an individual group form a UAV ad hoc network, which corresponds to the lowest layer of the multi-layer UAV ad hoc network architecture; the upper-layer UAV ad hoc network is composed of the backbone UAVs of all groups). They conclude that a decentralized UAV ad hoc network is the most appropriate architecture to connect a team of UAVs, while a multi-layer UAV ad hoc network is more suitable for multiple groups of heterogeneous UAVs.

Furthermore, when it comes to designing a safe and secure FANET routing protocols, only few of the FANET routing protocols proposed in [5] have been designed with an agile methodology. However, safety validation is necessary to enable the possible several UAS applications into a non-segregate airspace.

The existing work combining agile methodology and UAS mainly consist of designing the physical parts design of the UAVs. For instance, the authors in [6] have designed a UAV-dedicated GPS algorithm using an MDD approach. Moreover, in [7], MANET reactive protocols have been validated using MDD approach. Similarly, in [8][9], secure FANET routing protocols is designed with MDD approach while taking into account the FANET safety requirements. The authors use a general network architecture and perform unit testing and code verification through formal verification tools for safety validation purposes.

In the same way, a reference model architecture [1] is being proposed aiming at developing UAS focusing in certification.

## III. Experiments with a real FANET prototype

This section describes the methodology used to the development of FANETs using Model Driven Development (MDD) and presenting a real FANET prototype.

### A. Model driven development as a tool for real FANETs experiments

FANET safety must be validated to act as autonomous systems without a dedicated safety pilot and to be authorized to fly in the general airspace. Although specific validation and certification standards are yet to be proposed for FANET, the process and safety standards depicted within the DO 178C [10] are good candidates as explained in [11].

Our model is designed with Matlab Simulink and Stateflow frameworks. Simulink allows us to accurately design the routing protocol and Stateflow allows to define a finite number of states in the algorithm.

Figure 2 represents our model driven development workflow. It is composed of the seven different steps. In the first step, the specification is validated and partitioned into several subsets of the requirements. Each subset has its own unit test objectives. Then, in the second step, each partition is designed into a high-level model using graphical descriptive language provided by Simulink and stateflow tools. The network layer is divided into several blocks exchanging messages between them and adjacent layers. Each block is designed as a state machine which analyzes incoming requests, then sends them to each iteration.

Then, during the third step, each block model is converted into C library source code by Mathworks Embedded coder. It should be noted that these source codes are independent of any operating system or hardware architecture. The next step (called glueing) consists of linking the generated code to the kernel space of our target operating system (Embedded Linux generated from OpenEmbedded). In the fifth step, the library code and glueing code are combined into binary files. The next step consists of aggregating the previous binary files with our target operating system to provide a final binary image. The last step is the execution of binary image into the target hardware for verification and validation. The objective is to confront the routing protocol efficiency to a set of unexpected hazardous issues that come with the real environment.

### B. Real world experiment validation

To execute the source code obtained through MDD approach, it is necessary to cross-compile and cross link the embedded Linux C code into the target hardware, which is an ARM board. This cross-compilation is executed with OpenEmbedded framework [12]. Along with bitbake tool [13], OpenEmbedded allowed us to cross compile and merge the required software package. It also allowed us to generate, configure and build a lightweight and efficient Linux distribution for the ad hoc network communication.

*1) Experimental details:* The FANET architecture that has been set up is shown in Fig.3. The network is composed including 2 fixed wing UAVs (DT18 developed by the Delair Tech company) and one GCS (also from Delair Tech). The DT-18 is a lightweight UAV capable of long-range flight whose characteristics are detailed in Table I. The first UAV called Dr1 is flying at a distance of 250-500 meters from the GCS whereas the second UAV Dr2 flies at a distance 1500-2000 meters from the GCS. Hence, the distance between the two flying UAVs is approximately 1500m. The average altitude of UAVs is estimated at 1127 feet (345 m).

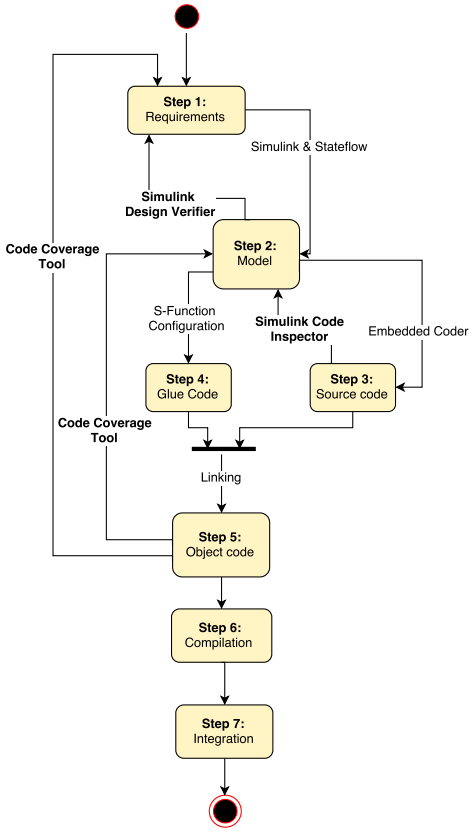



Fig. 2: Set of MDD tools used to design the secure FANET routing protocol

Furthermore, regarding the different types of traffics exchanged between UAV, there are 5 types of data traffics depicted in table II.

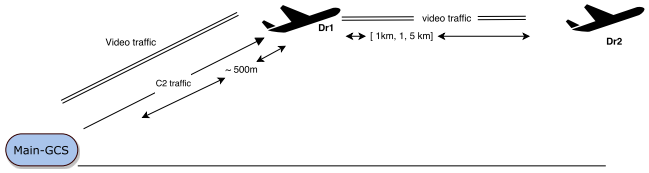



Fig. 3: FANET real world experiment architecture

| Characteristic | Value |
|---|---|
| Model | DT-18 |
| Payload | 250 g |
| Range | 100 km |
| Cruise speed | 50 km/h |
| Wind | up to 45 km/h |
| Real-time payload transmission | up to 15km. Extension to 100km |
| Autopilot | Delair-Tech technology |
| Onboard computer | payload and communication control, 1 Ghz |

TABLE I: Main characteristics of DT-18 UAVs

Moreover, on each UAV, we mount an ARM-based computer-on-module produced by Phytec Inc. It runs cus-

| Type of traffic | Source – Destination | Size | Flow rate |
|---|---|---|---|
| Heartbeat or Tick | GCS – Dr1<br>GCS – Dr2 | 64 Bytes | 1 packet/s |
| Geographical reference | Dr1 – GCS<br>Dr2 – GCS | 80 Bytes | 3 packets/s |
| C2 | GCS – Dr1<br>GCS – Dr2 | 80 Bytes | 1 packet/s |
| Video | Dr2 – Dr1 – GCS | 1400 Bytes | 25 UDP packets/second width=720,height=576 |
| Network | Exchanged between Dr1, Dr2 and the GCS | Request: 66 Bytes<br>Response: 62 Bytes<br>HELLO : 62 Bytes<br>Error: 54 Bytes | 1 packet/s for the HELLO Request and Response and Error packets are exchanged during disconnection (route loss) |

TABLE II: The different flows exchanged during the flight

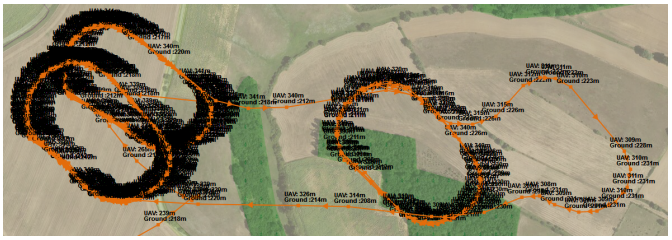


Fig. 5: UAVs mobility

tomized Linux distribution compiled and built with OpenEmbedded framework. We also attached a HD camera and a Wi-Fi radio interface module using IEEE 802.11n. During the experiments, we enabled space-time block coding to exploit channel diversity in order to avoid interference.The transmission bandwidth was set to 5 Mhz and provide half duplex communication. We used Quadrature Phase Shift Keying (QPSK) modulation techniques. Each UAV is remotely controlled through a dedicated GCS (Desktop and Infrastructure) as depicted in Fig. 4. However, they do not take part in the routing process. Their role is to provide safety link to each UAVs for failsafe management. This is part of DGAC[1] regulation to have at least each UAV connected to a station control via a dedicated link. Each UAV has therefore two links. The first link is either 868 MHz or 900 MHz obtained through XBee devices. The second link called safety link is a mandatory specification that must be set up to provide recovery assistance when the communication link through the XBee is not efficient. Fig. 4 illustrates the different types of data link deployed.

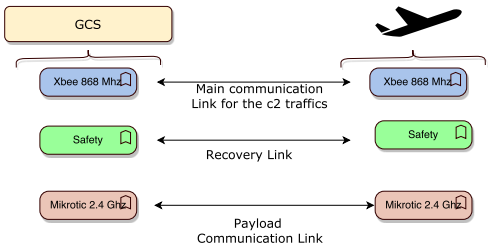



Fig. 4: Communication links between UVAs

Additionally, regarding UAVs mobility, it is a combination of rectilinear and circular movement remotely executed on demand by the operator through the GCS Desktop. The fluctuation between these mobility will create disconnection from time to time and decrease the performance accordingly. Fig. 5 illustrates the relative position of UAVs during experiment.

*2) Experiment results and analysis:* The experiment results are shown in Table III.

- The overhead represents the amount of control packet sizes added to data packets. Within FANETs, despite the constant mobility of UAVs, the signaling (size of network control packets) cost of reactive protocols is significantly lower than payload sizes. The protocol does not generate a significant amount of overhead. Thus, it does not perturb the transfer of payload (video) packets. This good overhead results can be explained by the non-requirements to acknowledge each packet sent compared to DSR protocol and the no need to update an already established route. Indeed, once a route between GCS and Dr2 (through DR1) is computed and established, the protocol no longer seeks the complete topology of the network. Each UAV only checks its direct neighbor through periodic exchanged of beacon messages.
- Route stability: Evaluates the delay during which the connectivity is uninterrupted. These results show that on average over 30 minutes test, there is a route loss each 14.5 seconds. This is explained by the UAV mobility degree that creates disconnections. Also, the difference in the radiation plane and the polarization plane of the transmitting and receiving antenna generates a signal attenuation which creates stability fluctuations. Moreover, apart the difference at the antenna level, it is important to mention that the introduction of real parameters in real situations impacts the performance. For example, it is noticed in the movement log that when the UAV is cornering, the inner wing mask the modem radio as it is located under the UAV. Generally, UAVs mobility and antenna polarization issues make it difficult to maintain communication for a long time.
- Average delay to re-establish route in case of route loss: This metric computes the recovery time to restore from route loss. This delay is equal to the time interval between the sender of route request (identified with a given sequence number) and the arrival of route response to restore a route. We can notice that the delay is quite low given the dynamic features of FANET. This is because of the small sizes of control packets and the speed of lights, which is approximately $3 * 10^8$ms - 1. Our protocol behaves correctly to its requirements specification. The topology change caused by UAVs mobility and speed does not affect significantly the delay to retrieve routes. This is also explained by the fact that the protocol does not wait for the loss of several beacon messages to start looking for an alternative route.
- Average end to end delay of control packets: It refers to the time taken for a packet to be transmitted across the network. It is composed of transmission delay,

[1]DGAC: Direction Gnrale de lAviation Civile which is equivalent to American FAA (Federal Aviation Administration)

| Metric | Value |
|---|---|
| Overhead Control packets | 352 ko |
| Overhead Traffic % (bytes) | 2.15 % |
| Average route lifetime | 14.33 s |
| Average delay to re-establish route | 0.001098 s |
| Average transmission delay between GCS and Dr1 | 0.000153 s |
| Average transmission delay between Dr1 and Dr2 | 0.000549 s |
| Connectivity during unstable states | 97 % |
| Number of control packet loss | 284 packets |
| Average size of payload packet loss during path loss | 52 Bytes |

TABLE III: FANET outdoor experiment results over 30 minutes

propagation delay, processing delay and queuing delay. We measured the transmission delay between Dr1 and GCS and between Dr1 and Dr2 to assess which part of the network take longer transmission. The two results indicate a good ability to forward control traffics within the network as we are noticing small delays. The routing protocol does not add significant further delays. Besides, the small sizes of control packets justify the result. We also notice that the communication delay of Dr1 and Dr2 is slightly more important than Dr1 and GCS. This is because the GCS does not move during the duration of the mission whereas Dr1 and Dr2 execute their respective flight plan. It is important to note that the delay may vary depending on the type of control packet. Nonetheless, due to the small size difference between these packets, we can ignore these differences.

- Connectivity during unstable states: we extracted disconnected states from the GCS to Dr2. To prevent short unstable states to disturb the measurement, two losses that are too close in time (less than 0.1s) are merged. Thus, we extracted the "unstable states, which corresponds to states when protocols do not provide a route as connectivity is partially halted. As we can see, the total connection time is better for the protocols. In unstable states, it stands out from the others making able connectivity up to 90%. This means that our protocol is reacting well to the topology changes, thus provide safe FANET communication.
- Loss Analysis of Control and Payload Packets: This metric measure how many routing control packets are lost during the mission. We also analyzed its impact on the payload traffics by measuring the size of payload packets being lost by the time the route is repaired. As shown in Table III, we notice a quite important amount of control packet loss (mostly the loss of HELLO packets). When a certain amount of HELLO packet is lost, there is a delay to establish a new route. This delay is relatively small but not negligible at this level. As a result, payload packets are fragmented and decreased in size. This indicates the degradation of the video quality viewed on the GCS desktop application during the mission.

## IV. FANETs SIMULATION

This section will present a FANET simulation carried out with OMNeT++. The OMNeT++ Discrete Event Simulator is a widely used network simulator that allows to visualize and analyse many network aspects. According to [14], the learning curve to use OMNeT++ is steep to start with. Thus, we have developed LARISSA, which is a layered architecture model that interconnects unmanned systems [1] and automatically generates code for OMNeT++. It helps with the process of creating OMNeT++ models for UAV domain applications.

### A. LARISSA as a tool for FANETs simulation

The term "architecture model" aims to incorporate the basic objective and system ideas. Among the advantages offered, we can mention: conceptual integrity, flexibility, reliability, improving the reusability and maintenance, higher level of abstraction and interoperability, more interactive interfaces between devices, certification of components and systems. In this sense it has been proposed a layered architecture called LARISSA: Layered Architecture Model for Interconnection of Systems in UAV.

So we implemented a profile in UML that defines a layered architecture for the UAS domain. Moreover, we implement $Model2Simulation$, which enables the generation of OMNeT configuration files, in order to perform data communication tests. Modeling a system using LARISSA Architecture provides a reliable approach since the components are already predefined in the profile. As well as modeling in UML, this approach transfers the designer a higher level of interaction with the project, since lower-level details were also implemented in the profile. The use of $Model2Simulation$ helps to easily and accurately generate parameters, as the specifications contained in the UML model.

*1) LARISSA Profile:* LARISSA profile was designed using the concepts of UML2 by OMG. The chosen tool for creating this UML profile was the Papyrus component from MDT (Model Development Tools) by Eclipse Foundation. Papyrus provides support for SysML (System Modeling Language), a general-purpose modeling language applied to engineering systems, supporting systems specification, analysis, design, verification and validation, including embedded systems and real-time systems.

LARISSA layers are represented in the form of packets and artifacts at the lowest level, represented by blocks extended from SysML. The properties of the blocks were represented as required and some of the possible values to be assigned were modeled using the Enumeration artifacts representation. An example of this model is shown in Fig. 6, in which the *AirToGround* sub-layer was modeled. It belongs to the *Communications* sub-layer, that belongs to the *AvionicsElectronics* sub-layer, which in turn belongs to the *Physical* layer under the Aerial segment.

The *TelecommandTelemetry* sub-layer implements the *RadioModem* stereotype, which is a generalization of the *WirelessDevice* stereotype.

### B. Experimental Design

The simulation experiments we use the trace file from the real experiments presented in section III, and we also increased the number of UAVs keeping just one GCS. The chosen parameters for the simulation are: 1) communication

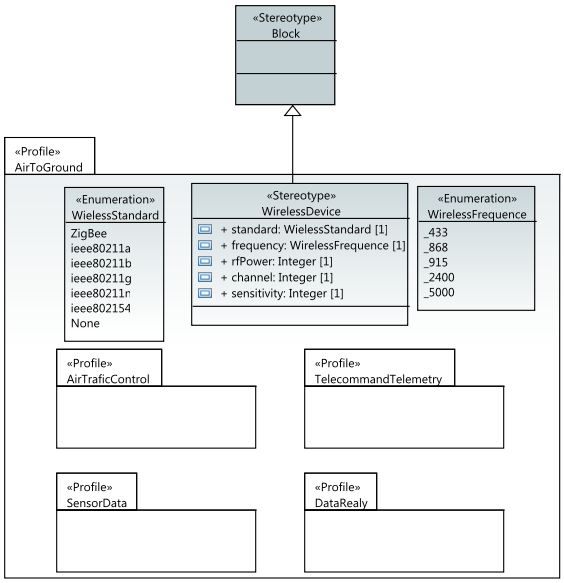



Fig. 6: Sub-layer *AirToGround*, from sub-layer *Communications*.

protocol: (a) IEEE 802.11n was chosen as a protocol due to its high use in UAVs data exchanges; (b) IEEE 802.15.4 was also chosen for the experiments and comparison due to the low cost, low power consumption, and high connectivity. 2) network topology: (a) Star was chosen once it is one of the most common topology in ad hoc networks (broadcast); (b) Mesh was chosen because of the mobility inherent in ad hoc networks, mainly in FANETs (AODV protocol). 3) amount of UAVs: we exponentially increase the amount of UAVs keeping just one GCS in each case (16, 32, 62,64 and 128 UAVs); 4) UAV speed: two different speed were chosen (low: 25m/s and high: 50m/s).

### C. Results and analysis

In our experiments, we have chosen four scenarios varying the number of hosts in order to cover smaller networks (like in [15]), and also bigger ones. The experiments were run with 16, 32, 64 and 128 hosts distributed in matrices $n \times m$ on OMNeT++ simulator, being $4 \times 4$, $5 \times 5$ (same positions with more than one host), $8 \times 8$ and $11 \times 11$ (same positions with more than one host), respectively. The distance among each position vertically and horizontally was fixed to 160 m. In all the experiments, as the number of hosts was increased, the bigger was the impact on network performance degradation, except for IEEE 802.11n with AODV routing protocol, which had a high rate of successfully transmitted packets.

The simulation time in all the experiments was set for 1000 seconds. It was chosen to provide a simulation time similar to the flight times observed in small UAVs. Thus, we run the experiments and compared the rate of successfully transmitted packets. Figs 7 and 8 show the results for IEEE 802.11n and IEEE 802.15.4, respectively.

The general behavior of IEEE 802.15.4 simulation can be described as a reduced fraction of successfully transmitted packets. The application of AODV routing protocol and the increasing of network hosts caused even worse results.

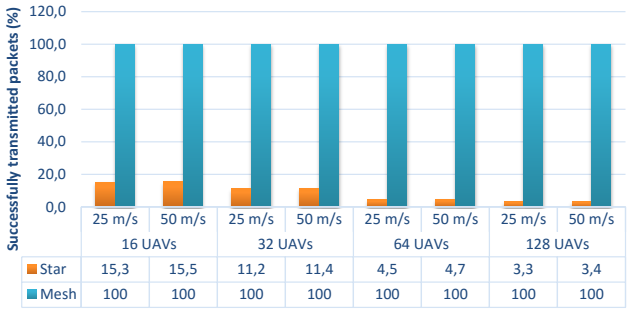



Fig. 7: Comparison of successfully transmitted packets by IEEE 802.11n.

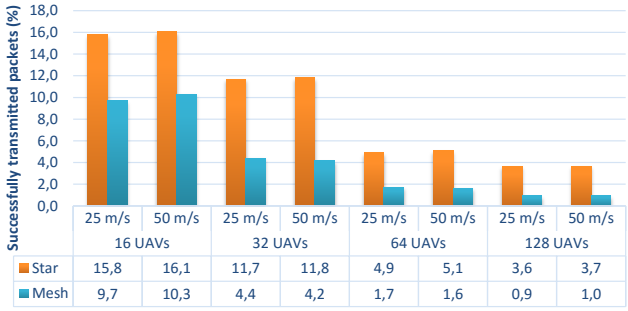



Fig. 8: Comparison of successfully transmitted packets by IEEE 802.15.4.

On the other hand, the general behavior of IEEE 802.11n changed considerably with AODV routing protocol. Without a routing protocol, the successfully transmitted packets rate is similar to the results observed for IEEE 802.15.4. However, with AODV routing protocol on IEEE 802.11n, we have noticed that almost 100% of packets successfully reached their destination. The number of hosts did not affect the performance of IEEE 802.11n with AODV routing protocol.

Fig. 9 presents the star configuration (without a routing protocol) and Fig. 10 presents the mesh configuration, which are the results with AODV routing protocol.

Furthermore, we have also analysed the end-to-end delay presented by each protocol with star and mesh configurations. The more UAVs join the FANET, the bigger is the delay in both cases for IEEE 802.11n protocol. However, on star configuration the speed is even more important on delay increasing, as it can be seen in Fig. 11. On the other hand, on mesh configuration there is no difference due to the speed change, as show in Fig. 12.

The end-to-end delay analysis for IEEE 802.15.4 presented a different behaviour. On star configuration, the main factor that increases the delay is the number of UAVs (Fig. 13). On the other hand, on mesh configuration the delay numbers are very high and did not change much because of the number of UAVs, if compared, as shown in Fig. 14.

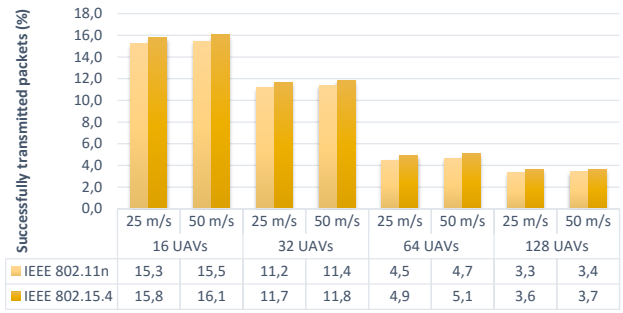



Fig. 9: Comparison of successfully transmitted packets in star configuration.

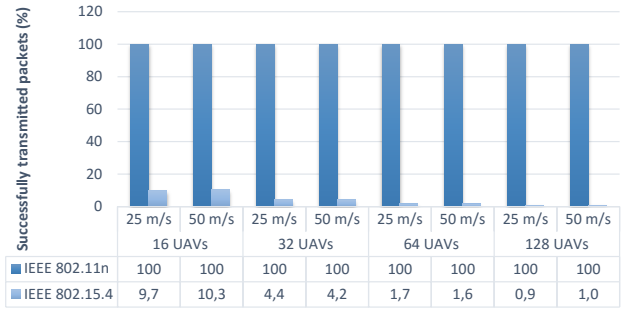



Fig. 10: Comparison of successfully transmitted packets in mesh configuration.

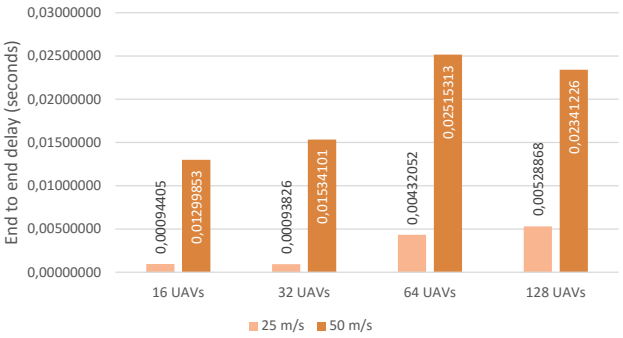



Fig. 11: Analysis of end to end delay on IEEE 802.11n with star configuration.




Fig. 12: Analysis of end to end delay on IEEE 802.11n with mesh configuration.




Fig. 13: Analysis of end to end delay on IEEE 802.15.4 with Star configuration.




Fig. 14: Analysis of end to end delay on IEEE 802.15.4 with Mesh configuration.

## V. Discussion

Based on real experiments (presented in section III) we can notice AODV provides a good performance with IEEE 802.11n protocol in the amount of transmitted packets even with packets loss (mainly HELLO messages). Comparing these results with the simulation (same scenario - 2 aircraft and 1 control ground station, as seen in section IV), we notice the same behavior, AODV protocol reaches high rate of successfully transmitted packets. We can assume, based on these results that the simulation behavior is similar as the number of UAVs is increased. So, we carried out simulations changing the amount of UAVs to assess the impact of topologies in safe FANETs.

The problem with IEEE 802.15.4 protocol is seen in a mesh topology due to the broadcast storm caused by the high amount of HELLO messages. Delay in the reconnection outcomes from the loss of HELLO messages when there are route losses

or UAV disconnection from the FANET. This can be solved using IEEE 802.11n protocol, which provides high delivery rates in mesh topology even with a huge number of UAVs.

Another way to solve this problem is to mitigate the HELLO messages since there is a storm broadcast problem taking place. It can be seen as a threat, once it might cause a non-intentional denial of service (DoS) attack. Solving this problem it will be possible to use IEEE 802.15.4 and IEEE 802.11n protocols in both topologies ensuring a low end to end delay, and high deliverable rates.

So, if an IEEE 802.11n device can connect directly to the Internet Protocol (IP) network, why even consider using IEEE 802.15.4 technology that requires an extra bridge once it is not IP-based? The reasons are cost, power consumption, and complexity [16]. IEEE 802.11n usually requires a higher-end micro-controller or microprocessor to avoid a bottleneck of messages in its traffic, so better processors are more costly. Another problem with IEEE 802.11n is the constant connection needed, to allow data to get through, consuming precious energy.

Once an IEEE 802.11n connection is a constant wireless link, more complex software is required to handle cases in which the connection is dropped. With IEEE 802.15.4 there is no connection that needs to stay open (the end device can just wake up, send its message, wait for an acknowledgment, and then go back to sleep), allowing the device to transmit at higher power levels (longer range) and save more power by spending less time with an active Radio Frequency connection.

One can consider using these protocols, and face issues such as power consumption, cost and complexity (UAVs biggest concerns), IEEE 802.15.4 would be considered as the best choice.

## VI. Conclusions

In this paper we have analysed FANETs in real and simulated environments aiming at knowing how to provide safe ad hoc networks. Thus, we have carried out experiments showing the behavior of IEEE 802.11n and IEEE 802.15.4 operating in star and mesh topologies. The simulation results show that star network topology is affected by high UAV density and speed, which impact negatively in the packet delivery rate and the end to end delay.

These results demonstrate that within star topology more network resources are used, thus collisions occur. Also due to the high speed of the UAVs, the dedicated link between each UAV and GCS fluctuates and affects data exchanges. We can conclude that FANETs using mesh topology with IEEE 802.11n are safer than using star topology with the same protocol. Although the performance of IEEE 802.15.4 was not as better as IEEE 802.11n in mesh topology, it should be considered with mesh topology and as a subject of a future study due to the low cost, low power consumption and high connectivity.

An automatic OMNeT++ code generation using a reference model architecture (LARISSA), which guarantees the real modeling and standard information about the aircraft, has been provided. In the same way, MDD has been used to model and test routing algorithms. The performance results show that the routing protocol fits well to the dynamic topology of FANET. Despite the route loss for every 14.5 seconds which degrades the video traffic quality, we noticed that the delay to repair routes is relatively small. However, this should be improved to have more stable routes.

Regarding our short-term perspective, we would like to improve the routing protocol by adding security, mitigating messages that can lead to DoS attacks, and perform additional real-world tests to validate security and safety aspects of FANETs.

## Acknowledgment

The authors would like to thank the financial support granted by FAPESP (process number 2012/16171-6).

## References

[1] E. A. Marconato, D. F. Pigatto, K. R. L. J. C. Branco, and L. H. C. Branco, "Larissa: Layered architecture model for interconnection of systems in uas," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, May 2014, pp. 20–31.

[2] R. S. Hanmer, D. T. McBride, and V. B. Mendiratta, "Comparing reliability and security: Concepts, requirements, and techniques," *Bell Labs Technical Journal*, vol. 12, no. 3, pp. 65–78, 2007.

[3] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (fanets): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.

[4] J. Li, Y. Zhou, and L. Lamont, "Communication architectures and protocols for networking unmanned aerial vehicles," in *2013 IEEE Globecom Workshops (GC Wkshps)*. IEEE, dec 2013, pp. 1415–1420.

[5] O. K. Sahingoz, "Mobile networking with uavs: opportunities and challenges," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013, pp. 933–941.

[6] D. Santamaría, F. Alarcón, A. Jiménez, A. Viguria, M. Béjar, and A. Ollero, "Model-based design, development and validation for uas critical software," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 103–114, 2012.

[7] A. Geraldy, R. Gotzhein, and C. Heidinger, "Model-driven development of complex routing protocols with sdl-mdd," in *Joint ITU-T and SDL Forum Society workshop on ITU System Design Languages, Geneva, Switzerland*, 2008.

[8] J.-A. Maxa, M. Slim Ben Mahmoud, and N. Larrieu, "Secure routing protocol design for uav ad hoc networks," in *Digital Avionics Systems Conference (DASC), 2015 IEEE/AIAA 34th*. IEEE, 2015, pp. 4A5–1.

[9] J.-A. Maxa, M.-S. B. Mahmoud, and N. Larrieu, "Joint model-driven design and real experiment-based validation for a secure uav ad hoc network routing protocol," in *ICNS 2016, 2016 Integrated Communications Navigation and Surveillance Conference*, 2016.

[10] Y. Moy, E. Ledinot, H. Delseny, V. Wiels, and B. Monate, "Testing or formal verification: Do-178c alternatives and industrial experience," *Software, IEEE*, vol. 30, no. 3, pp. 50–57, 2013.

[11] N. Larrieu, "How can model driven development approaches improve the certification process for uas?" in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE, 2014, pp. 253–260.

[12] O. Team, "Openembedded user manual," 2006.

[13] M. Lauer, "Building embedded linux distributions with bitbake and openembedded," in *Proceedings of the Free and Open Source Software Developers European Meeting (FOSDEM), Brussels, Belgium*, 2005.

[14] T. Chamberlain, *Learning OMNeT++*. Packt Publishing Ltd, 2013.

[15] K. Singh and A. K. Verma, "Experimental analysis of aodv, dsdv and olsr routing protocol for flying adhoc networks (fanets)," in *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–4.

[16] K. Furlong and R. Erickson, "The Power of 802.15.4 and Ethernet," Tech. Rep., 2011. [Online]. Available: https://www.lsr.com/white-papers/the-power-of-802-15-4-and-ethernet