IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 15, 2019, accepted October 17, 2019, date of publication October 23, 2019,
date of current version November 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949119

VENUE: Virtualized Environment for Multi-UAV Network Emulation

VICTOR SANCHEZ-AGUERO [1,2], FRANCISCO VALERA [2], BORJA NOGALES [2], LUIS F. GONZALEZ [2], AND IVAN VIDAL [2]
[1]IMDEA Networks Institute, 28918 Madrid, Spain
[2]Telematic Engineering Department, Universidad Carlos III de Madrid, 28911 Leganès, Spain

Corresponding author: Victor Sanchez-Aguero (victor.sanchez@imdea.org)

This work was supported in part by the 5G-City Project under Grant TEC2016-76795-C6-3-R through the Spanish Ministry of Economy and Competitiveness, and in part by the H2020 5GRANGE Project under Grant 777137.

ABSTRACT Unmanned Aerial Vehicles (UAVs) have progressively been integrated into people lives during the last years. It is quite common now to see UAVs flying in the countryside doing field inspection, in highways for traffic control operations, or above stadiums in sport and music events. It is also common to see spectacular UAV swarm showcases (in most cases they are just performing a choreography) showing the potential of upcoming technologies. This article is focused on multi-UAV scenarios, on the establishment of Flying Ad hoc Networks (FANETs), and on the integration of 5G technologies like Network Function Virtualization (NFV) or Software Defined Networking (SDN). In particular, this article presents a proposal for one of the most common problems that the research and development community has to face at some stage: the validation of the different solutions and deployments. In this area, there is currently a notorious gap between the design phase and the deployment phase, since traditional network simulators are not designed with the constraints imposed by UAVs in mind. Besides, services implementations (that are usually distributed into single-board computers carried as payloads by UAVs) cannot be easily combined with the simulators. VENUE (Virtualized Environment for multi-UAV network emulation) is presented as an experimentation platform that allows testing the integration of multi-UAV FANETs together with network services deployments. VENUE covers from the simulation/emulation phase up to the real equipment integration phase. The validation of the platform is also presented in this article through several UAV use cases that make use of NFV technologies.

INDEX TERMS Emulation, FANETs, LXC, multi-UAV, ns-3, OLSR, SBC, swarm, SUAV, UAV.

I. INTRODUCTION

According to the Aerospace Forecast report provided in [1] by the U.S. Federal Aviation, nowadays, most commercial UAV missions are devoted to research and development or training and education. Despite the optimistic commercial predictions included in this report, the sector still awaits for a consistent Unmanned Aircraft System Traffic Management regulation [2], [3] to safely enable multiple UAV commercial operations beyond visual line-of-sight. To be ready for this near-future scenario, the scientific community is focused on the development of new solutions.

Traditional UAV services are mostly inherited from modelers UAVs, based on video surveillance or single-UAV use cases like precision agriculture [4], emergency response, healthcare, video surveillance for wildfire [5], journalism [6], disaster management [5], data collection in sensor networks [7], extended access network [8] or traffic monitoring [9], [10]. However, many research challenges are mainly focused on multi-UAV service provisioning (the so-called UAV swarm) whose communications are intrinsically related to the different Flying Ad hoc Networks (FANET) proposals.

It is in these scenarios where the fast evolution in the UAV research area, in combination with the improvements in the miniaturization of electronics and sensors, brings to the scene the use of Small UAV (SUAV). SUAVs can be produced at a low cost, and their reduced size enables the chance of their coordination and collaboration to provide flexible environments where network services can be distributed not only from the CPU or memory resource perspective but also from the geographical viewpoint. In general, multi-UAV systems

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

are scalable, flexible, and since the possibility of failure is distributed, they can provide service survivability [11].

However, multi-UAV systems do present some challenges that must be solved for suitable performance. Some of these essential design challenges are associated with wireless communications because of the particular characteristics of multi-UAV networks [12]. These particularities are: (i) the mobility degree and mobility pattern (different from popular mobile wireless networks) since it can lead to intermittent connectivity states; (ii) the dynamic network topology (it changes depending on the target mission) because device to device communications must be maintained despite the changes in the topology; (iii) SUAV networks include different types of sensors (requiring for diverse data delivery strategy, e.g., network paths, priorities).

The environment for service provisioning that multi-UAV systems are showing is a particularly appropriate context for the 5G softwarization technologies such as the Network Function Virtualization (NFV) (enabling a faster and more flexible deployment of network services), or Software Defined Networking (SDN) (enabling an easier network configuration and reducing the operational costs). It is still an emerging research area, but there are already different articles showing the potential and also the challenges of these 5G solutions [13]–[16].

In this scenario, one of the most prominent challenges that still must be faced is the enormous existing gap between all these new solutions and their deployment in real scenarios, since field tests are difficult (complex, but also quite restrictive due to regulations) and expensive to perform, and simulation alternatives are not appropriate because they have not been specifically designed for these use cases. Consequently, to address the inherent challenges of multi-UAV systems and solutions, an intermediate step between the design and the real validation process is required.

Taking into account the aforementioned considerations, this article presents an open-source validation platform for multi-UAV and FANET scenarios where different services based on 5G programmable UAVs can be deployed and tested. This solution is built on top of Linux Containers (LXC) and the ns-3 network simulator (based on [17]), and provides an emulation framework that allows the integration of different Virtual Network Functions (VNFs) (or virtual entities in general), which can be later used into real SUAV hardware, together with a network simulator (used to emulate the specific characteristics of wireless channels).

Besides, the framework enables real hardware (that can be on-boarded into SUAVs as payload) to be directly integrated with the simulation environment, in order to not only test the performance of applications and developments but also to test the correct operation in the hardware that will host the developments in the real world.

In conclusion, this framework facilitates the prototyping and validation processes of multi-UAV services and provides an ecosystem to test the developments that will later be used on real infrastructures (in particular, UAV equipment). The goal of this article is not to evaluate a specific routing protocol or any particular development but to describe VENUE and to exemplify its possibilities to perform the evaluations desired by its users.

The rest of the article is organized as follows: in section II, the related work and background are reviewed. Section III details the development of the network emulator. Section IV presents a use case where the platform is used to evaluate routing protocols for FANETs. Section V presents an NFV use case where a multi-UAV network is utilized to increase the programmable network resources over a delimited geographic area, enabling the communications between two groups of users. Finally, section VI concludes the article.

II. RELATED WORK AND BACKGROUND

In this section, (i) FANETs and some of their challenges are introduced, together with (ii) the most used network simulators to validate these networks. The section also includes (iii) the current situation of specific developments in simulators for UAVs and finally (iv) the contributions of the platform in comparison to existing developments.

A. FANETS

Motivated by the emerging development of SUAVs, the FANET [12] concept has been increasingly attracting the attention of both military and civil environments because of its simplicity, versatility, flexibility, and usability. FANETs are ad hoc networks built by aerial mobile nodes, and one of their most remarkable advantages is to be able to provide feasible Device to Device (D2D) wireless communications between network nodes without any need for additional infrastructure, like traditional Mobile Ad Hoc Networks (MANETs) or Vehicular Ad Hoc Networks (VANETs). Previous research in the area of MANETs and VANET has indeed served as FANETs starting point, although it cannot be directly applied because MANETs and VANETs have been designed for devices with limited speed or restricted mobility patterns, and multi-UAV networks are in general, very dynamic in nature. Therefore, it is required to study FANETs as a new network family due to their different requirements regarding QoS, mobility models, or data delivery [11].

Before a multi-UAV system can provide stable and reliable services, there are diverse challenges to be considered, and one of the most relevant ones is related to the number of nodes and links that coexist and cooperate in the network. UAVs do regularly change their position (at least for battery replacement), and these movements do obviously modify the network topology and so the connections created among the nodes. A simple change in the location of a UAV that happens to be connecting different parts of the network may lead to a topology partition and service performance degradation. The same may happen when a UAV has to be replaced (because of battery exhaustion, for instance) forcing the new incoming

UAV to have proactive/reactive restoration mechanisms to be properly configured.

B. NETWORK SIMULATORS

Addressing these challenges and many others (for example, to evaluate the performance of routing algorithms, to examine and to validate mobility models, or to understand the behavior of each node in the network [11]) is a hard and expensive process because it is difficult to verify the different proposals in a real (flying) environment. On this basis, multi-UAV network solutions are commonly trialed using network simulators. The research community provides multiple alternatives for network simulation and experimental validation, such as the Network Simulator 2 (ns-2) [18], the Network Simulator 3 (ns-3) [19], OMNET++ [20] or Mininet [21]. Ns-2 provides substantial support for TCP simulation, routing, and multicast protocols over wired and wireless (local and satellite) networks. ns-3 is also an open-source discrete-event network simulator which primarily targets for research and educational use. OMNET++ is an extensible, modular, component-based C++ simulation library and framework, mainly intended to build specific network simulators. Mininet is used to create realistic virtual networks, running real operating system kernels, switches, and implemented application code, on a single machine. Although there is a remarkable interest on these validation platforms, and a large community of users following and evolving them, in certain particular scenarios, like the ones enabled by multi-UAV networks, it is difficult to have realistic results using these well-known network simulators because more specific modules are still required. For example, beyond the evaluation of the communication channel that may be reasonably modeled by existing simulators, the analysis of new services and applications that run in the UAV equipment, e.g., novel FANET routing protocols, innovative sensor information distribution protocols, or new 5G softwarization technologies like SDN or NFV, requires a considerable improvement in current simulators to be able to run new service logics on top of them. Moreover, there are still different areas where additional developments would significantly improve the application of simulators for UAVs. For instance: to enable the interaction of real hardware with the simulated nodes, to add realistic energy consumption models based on measurements, the integration of the simulated network with a 5G core, etc.

As a result, in the state of the art, there are numerous examples of network simulator expansions to satisfy particular situations (even for issues not related to UAVs). NEMAN [22] extends ns-2 to allow running a virtual wireless network of hundreds of nodes on a single end-user machine. Dockemu [23] extends ns-3 to provide a flexible system to rapidly create networks (wired or wireless), incorporating the latest developments and a user-friendly method of installation and configuration. TapRouter [24] presents an application-emulating framework for MANETs with high performance and usability by integrating the ns-3 and lightweight virtualization technology.

C. UAV SIMULATOR

There are, in fact, specific multi-UAV simulation platforms, but they are mainly focused on flight operations (e.g., landing, refueling, mobility) or focused on applications on top of the multi-UAV system that are not related with network communication scenarios. RotorS [25] is a modular Micro Aerial Vehicle framework based on Gazebo [26] targeted to tackle higher-level tasks, such as collision avoidance, path planning, and vision-based problems, like Simultaneous Localization and Mapping (SLAM). OpenUAV [27] is a development to avoid the high barrier imposed by the use of flight UAV simulations due to the need for powerful computers and the time required for the initial set up. There are also several examples of papers focused on flight tasks [28].

Some other works that do consider UAV communications are, in general, evaluating their performance directly using general-purpose network simulators [29]–[31], assuming significant simplifications, e.g., outdated mobility models, limited simulation times, or limited traffic patterns.

D. VENUE CONTRIBUTIONS

VENUE allows testing a wide variety of scenarios including protocols, services, and technologies embedded in the on-board computer of the UAVs. This includes FANETs communication technologies based on standard protocols such as OLSR, AODV or BATMAN, or based on new trends such as SDN. This contribution also includes the possibility of testing from traditional services and applications such as voice over IP or video streaming, to innovative services based on virtualization or 5G technologies such as NFV.

Another of VENUE significant contribution is the possibility to interact with real hardware that is frequently used as a UAV payload. The real payload allows not only to test the development functionality but also to test if the hardware has adequate resources. Modifications had to be made to the ns-3 source code to allow the interaction on different entities (virtualized or real hardware), as this functionality is not entirely supported.

The platform also enables to test scenarios that require network nodes mobility. In this respect, it supports Mission Planned Based (MPB) mobility pattern [32], i.e., predetermined trajectory information, which is usually planned in advance. This way, UAV mobility patterns can be specified by the platform user following a predetermined format described in [33]. The platform supports the installation of a predefined mobility pattern for each of the UAVs in the emulation. Therefore, it opens the possibility to use any of the mobility models defined in [34]. Each UAV follows MPB information with realistic flight traces.

Finally, the platform incorporates a number of pre-created modules that enable scenarios with the utilization of different routing protocols (e.g., OLSR, SDN based routing solutions), as well as a set of supporting tools to test user applications and developments. These modules can be flexibly incorporated into user-defined test scenarios, as they have been implemented using virtualization containers.

III. VENUE DESIGN AND IMPLEMENTATION

VENUE has specifically been designed to satisfy different aspects that are not fully supported in current network validation solutions (simulators/emulators), and that is important to be able to validate current FANET related technologies.

As it is detailed in this section, the most relevant requirements that have guided VENUE design are (i) The necessity to provide mobility models to accommodate real UAV applications and that are undoubtedly important since they define the FANET physical topology evolution during the service time (VENUE can introduce two-dimensional (x,y axes) mobility patterns in a predefined altitude (z-axis)). Second, (ii) the necessity not only to simulate services but also to be able to emulate them, testing the real applications that can later be migrated into real hardware. VENUE can serve as an emulator of the physical channel linking real applications and processing real packets (the platform can model and select the communication channels and technologies between the different participants of the FANET). This feature also enables VENUE to serve as a validation platform until almost the final integration with the hardware devices where the services will be installed (in fact, this hardware can also be directly attached to the platform to validate this phase too as it will be seen). Finally, (iii) the necessity to support 5G softwarization technologies, such as NFV and SDN [35]–[37]. These technologies are being introduced into the UAVs field as innovative alternatives to be able to support flexible (and agile) service provisioning (with NFV) but also as a possibility to evolve current ad hoc networks routing protocols adapting them to the particularities of FANETs (with SDN). VENUE improves the support provided in ns-3 to be able to integrate multiple external nodes into ad hoc networks emulating the communication channel. These external nodes can incorporate the corresponding network functions into the system and test them all together.

One of the main strengths is that the framework operates with real applications, e.g., Linux Containers, Virtual Machines, real hardware, that can be directly used in real infrastructure afterward (consequently, VENUE is suitable for both IPv4 and IPv6). This functionality allows reducing prototyping and validation cycles and reducing the time-to-market or time-to-operation period. The framework is also suitable to emulate both wired and wireless (infrastructure-based and ad hoc) networks. Since this implementation is designed to trial FANET scenarios, all the examples provided in the framework are based on the 802.11 Wi-Fi technology. Still, with small effort, this platform may also assist in wired network scenarios.

The platform also enables more than one real host (virtual or physical) to interact with the network emulation because of the source code modifications (available in the patch file [33]). This functionality is crucial since there are usually numerous participants in a FANET. To facilitate the prototyping of all this potential participants, the framework incorporates pre-created Linux Containers that include the installation and configuration of different FANET routing protocols, such as the Optimized Link State Routing (OLSR) [38], Software Defined Networking (SDN) technologies (like RYU [39] controller and OVS [40]), and network analysis tools (iPerf [41] and Trafic [9]). In combination, the ns-3 simulator is installed and configured inside a Linux Container. Thus, the emulator can be used as a standard virtual function, with all the advantages it brings (the platform can be instantiated by any Virtual Infrastructure Manager (VIM) such as OpenStack and is portable to any Linux machine without any specific configuration). The platform also incorporates some scripts that configure the emulation environment for a smooth development process. This functionality allows the user to create complex multi-UAV networks in a simple way enabling advanced experiments to be carried out. To analyze and measure the service performances, ns-3 generates standard network traffic traces [42] that assist the process of code debugging and traffic analysis. The results can be studied using regular tools like Wireshark [43], which is utilized for network troubleshooting, analysis and also allows us to analyze the traffic that passes through a network and thus can solve or even prevent possible problems that may arise.

Finally, by default, the framework incorporates the Mission Planned Based (MPB) mobility model [32]. The platform user can introduce real flight traces (VENUE works with two dimensional traces, but the analysis is considered to be correct since landing and take-off are made far from the network service area and in the network service area in most situations it is reasonable to consider the UAVs flying in the same plane) following a predefined format to enable realistic multi-UAV mission.

A. FRAMEWORK ARCHITECTURE

Figure 1 summarizes the architecture of VENUE whose components are described in the following subsection. The global view of the system (complete architecture) is shown in the right part of the picture. The emulation layer is in charge of creating and modeling the FANET, and it also emulates UAV mobility. The top layer represents the real nodes (virtual entities and general-purpose hardware). Those nodes contain the developments and applications that enable the multi-UAV scenario (routing protocols, network services, data collection/transmission, etc.). The integration of the two layers not only allows linking real nodes through an emulated FANET but also provides each UAV with mobility. In the left part of the figure (node view), the connection between the ns-3 nodes and the real nodes is highlighted. Moreover, it includes all the required components to make the association possible such as Linux Bridges (software used to join two or more networks that behave like a virtual network switch) or TAP interfaces (network interface entirely supported in software) [17].

To implement the emulation layer, the ns-3 network simulator has been selected. ns-3 is a discrete event-based network simulator frequently used in the investigation of ad

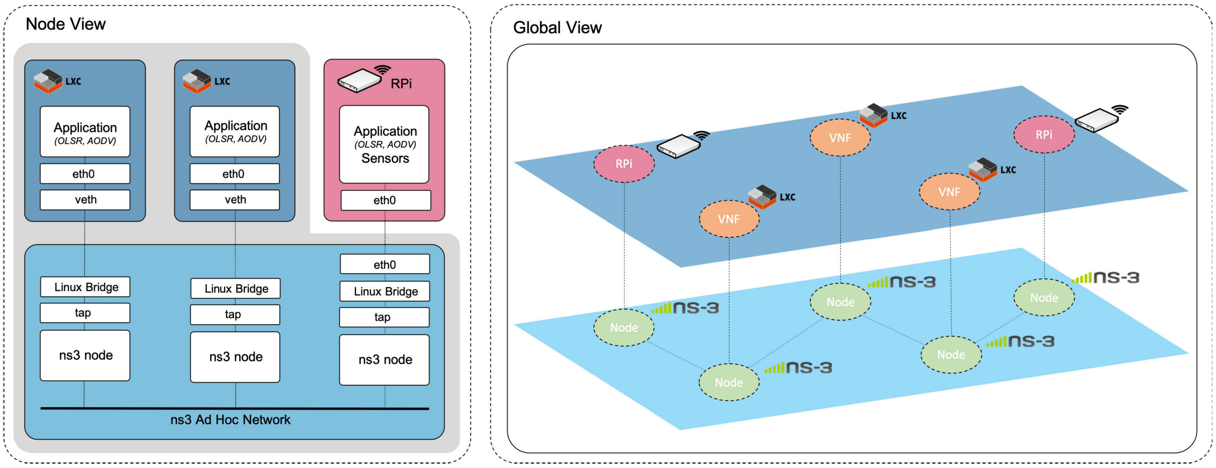


FIGURE 1. **Framework architecture, node and global view.**

hoc mobile networks. It implements a wide variety of routing protocols from both wired and wireless networks. ns-3 also allows the configuration of several network parameters; meanwhile, it provides extensive data collection modules for exhaustive analysis. However, the main reason to choose ns-3 is that in addition to its simulation capabilities, it implements a module that allows network emulation, i.e., this module allows external entities (real or virtual) to interact with the ns-3 environment. The emulation module provides a fully real-time controllable and reproducible environment where the routing protocols and application developments can later be used in UAV equipment without modifications. This functionality introduces a notorious added value as compared to pure ns-3 simulations since the ns-3 developments cannot be used in real equipment.

In FANETs, the number of participant nodes may be high, and in order to be able to validate these scenarios beyond the integration of real devices (to test real hardware inside a FANET with numerous participants), VENUE must be able to handle inputs from multiple virtual devices allowing that way to provide a scalable solution. Thus, the experimentation is not only limited by the availability of physical devices (which in these cases are usually expensive) but also by the selected virtualization technology that must be as lightweight as possible aiming to run several singular applications (each of the SUAVs inside the multi-UAV system) on a single server. For this purpose, the Linux Containers (LXC) [44] have been selected. However, LXC is the tool that VENUE provides to ease prototyping; the developed application could then be used on any virtualization platform or directly on the physical UAV device. In Figure 1 (Node view), it can be seen that all the required applications to enable communications between the UAVs (e.g., OLSR, SDN) are in the LXC (or in real hardware). This feature allows the tested applications to be portable to commodity equipment.

A Linux Container is a set of processes that are detached from the rest of the Operating System. Unlike standard Virtual Machines (VM), Linux Containers share the kernel with the operating system and separate the application from the rest of the system. LXC is portable and modular, including in the production stage (for instance, the same Linux Container can run in different hosts using an NFV platform). These characteristics make the prototyping and development process faster in comparison with traditional test environments. Nevertheless, the containers must be compatible with the underlying operating system (because LXC share the kernel). The selected hypervisor to manage the Linux Containers is the Linux Daemon (LXD) [45] because of its simplicity. LXD adds new possibilities and functionalities compared to the conventional system container management of LXC, e.g., container migration or physical devices passthrough.

These Linux Containers have proven to be usable in UAV regular payload equipment (see our previous work in [46], [47] where the design of the solution is based on Network Function Virtualization (NFV) and lightweight Virtual Network Functions (VNF)).

Although one of the most significant strengths of the platform is the possibility of virtualization, this framework also allows the interaction with real hardware, allowing VENUE users to get an insight about the behavior of real hardware during a mission. Typically, UAV embedded equipment is reduced in size and limited in both computing capacity and battery. Therefore, although the developed application may have the correct functionality, it cannot be guaranteed that the hardware in charge of its execution will have enough resources. Thanks to this integration, the limitations of the multi-UAVs payload hardware can be estimated.

To integrate real hosts (including either real hardware, Raspberry Pi (RPi) in Figure 1, or virtual hosts) into the ns-3 emulation, it is necessary to use the TapBridge Model [48]. This module allows the replacement of specific nodes (previously determined by the user) from the ns-3 network by real hosts. The TapBridge Model overwrites the ns-3-device MAC address by the overlying real-host (virtual entity or UAV payload) MAC address. After this association, the real-host considers the ns-3 net device as a local device and the TapBridge Model sends all the ns-3 node incoming network traffic through a virtual TAP interface (which is connected to the LXC container or the UAV equipment through a Linux Bridge as it is shown in Figure 1). Similarly, the Tap Bridge Model sends all the outgoing traffic (virtual entity or UAV payload) through the emulated ad hoc network. Thus, real devices can communicate with each other using the underlying network created by the ns-3, as can be seen in Figure 1.

TapBridge uses an existing TAP interface previously created and configured by the user. Nonetheless, in VENUE, the process of creating and configuring the system environment has been automated, and the user only needs to provide some input parameters, according to the experimentation scenario. More details can be found in [17] and [33].

However, the default TapBridge device presents problems to perform more than one "<ns-3 - real host>" MAC association in wireless networks (to connect more than one real device to the emulated network). For VENUE, it has been required to apply some modifications to the source code of the ns-3 to make the connection between several real hosts and several ns-3 nodes. These bugs have been reported [37] and are under revision. However, in the meantime, a patch file with the corrections is provided to apply those changes in [33].

IV. USE CASE I: ROUTING PROTOCOLS FOR FANETS

The first use case presents a simple FANET scenario in order to show how it is possible to validate the integration of real software (virtual entities in this particular case) with the VENUE platform, which is in charge of emulating not only the communication channel (Wi-Fi) but also the mobility pattern of each UAV. More specifically, in this testbed, VENUE is used to evaluate routing protocols in a FANET scenario. Following this methodology and using similar metrics to the ones proposed in this scenario, VENUE users can, for instance, select the most suitable routing solution for their own FANET scenario/mission or identify at a glance the most relevant UAVs inside the FANET from the communications perspective.

As has been mentioned, the goal of this use case is not to evaluate a specific routing algorithm. Therefore, the same analysis can be replicated with any other routing protocol selected by the user of the VENUE. For this use case, we have selected OLSR as an example because it is one of the most popular and well-known protocols in MANETs.

A. SCENARIO MOTIVATION

Multi-UAV systems may present heterogeneous mobility patterns, involving from a slow-changing topology to a dynamic and fast-changing topology. In fact, it is common to find challenges such as nodes with a high mobility rate (depending on the mission nature), damaged links, or battery constraints. Consequently, changes in the network topology are frequent in comparison to current mobile wireless networks. Moreover, environmental factors can significantly affect a FANET. Meteorological agents, e.g., wind gusts, rainfalls, high/low temperatures, resemble essential in the proper functioning of the system and cannot be accurately predicted before the beginning of the mission. Thus, the routing protocols algorithms require to go beyond the needs of usual MANETs and VANETs, and in consequence, the proposed solution must be flexible enough to allow different levels of dynamism.

Developing an autonomous and cooperative FANET requires reliable and robust communications between UAVs, which must collaborate to efficiently accomplish missions. To determine and configure the potential multi-hop network paths across the UAVs swarm, a routing/forwarding algorithm is needed. In these scenarios, data from each UAV can be sent through a possible connection with external infrastructure, that may act as a relay node, such as a Ground Control Station (GCS) or a Satellite link. Long-range communications facilities, e.g., Satellite or Line-of-Sight radio, imply heavy payloads and consequently flight restrictions and battery consumption. Another alternative is to directly send data through the FANET and eventually use a GCS to communicate towards other UAVs or ground infrastructure beyond the FANET.

B. SCENARIO DESCRIPTION

This scenario represents a service provided by a seven SUAV fleet that is used to enable communications, e.g., VoIP calls, video broadcasting, 5G network access, when the conventional cellular network is not available or is insufficient, e.g., emergencies, massified events. In this kind of FANET mission, SUAVs usually do have a fixed position and in fact, are perched on land wherever possible to save battery. Topological changes in these scenarios are typically generated because of battery constraints that imply SUAVs replacement. In this scenario, we consider that when a SUAV battery is under a determined threshold, the SUAV flies back to the GCS to charge/substitute batteries while another SUAV comes onto the swarm to provide the service of the replaced SUAV. This phenomenon affects the performance of the FANET because the rest of the devices should be updated to continue providing the network service. For this purpose, SUAVs must incorporate autonomously reconfigurable routing solutions that collect information about the current status of the FANET and configure the network paths autonomously.

The SUAVs are placed in a grid (50 meters between each UAV) with a static position (hoovering), as represented in Figure 2, forming a FANET. Each SUAVs is represented with a Linux Container using the VENUE platform. To measure the network performance, the source (SUAV #1) and destination (SUAV #7) Linux Containers have iPerf [41] installed. iPerf is a tool intended for active measurements that reports the bandwidth, loss, and other parameters. A traffic flow is sent from the source to the destination representing a voice over IP call (64 Kbits/s of data rate and 126 bytes of

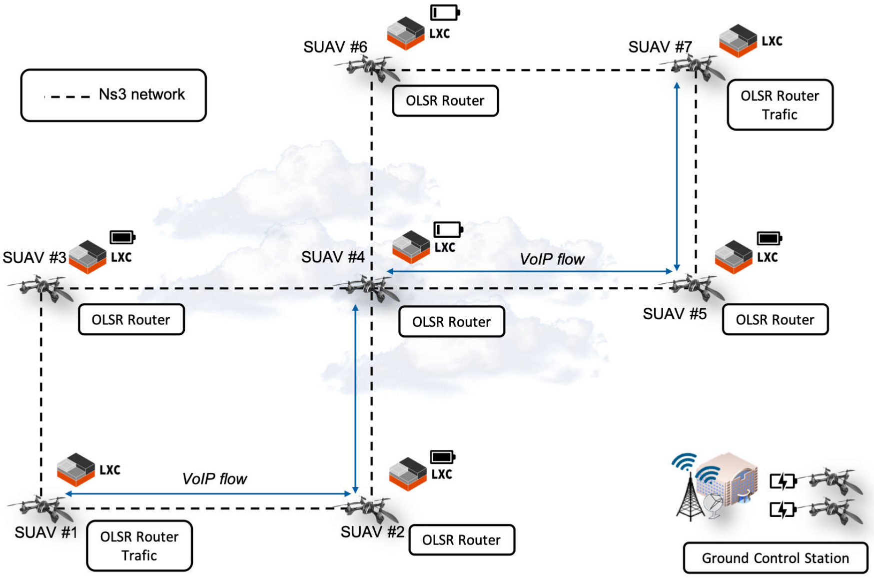


FIGURE 2. Scenario I: Routing protocols for FANETs.

packet length). The established ad hoc Wi-Fi network follows the 802.11a standard.

It has been selected a simple use case (in this case with only 7 UAVs) that allows understanding the functionality of VENUE without complicating the configuration and description of the scenario itself. The decision of placing the UAVs on a grid is intended to provide UAVs with more than one path to reach the possible destination. However, there is a central node through which all communications must pass, to illustrate in the analysis that the design of the network topology can significantly affect the service performance.

Typically, an SUAV battery has an autonomy of around 20 flight minutes (in similar cases where the SUAVs are landed to save battery, the lifetime of each device is longer). Taking into account the response time of regular routing protocols this replacement time is enough so as to allow a proper operation of the network (as it is shown in [46]), so in order to see how VENUE can operate, in this experiment we will increase the failure frequency to an extent that the FANET routing will be really disturbed by these failures (and consequently data moving through the FANET would also be disturbed). We have selected (unreal) battery lifetimes from 10 to 400 seconds to force several changes per minute. The main objective of this article motivates the selected replacement values (despite being unreasonable values beforehand in a real scenario). The primary purpose is to show how VENUE can be used, how to understand the results, and to propose a possible analysis). The primary purpose is to prove how VENUE works, how to understand the results, and to show the possible analysis. The battery lifetime is modeled for each UAV following a uniform distribution with a variation of ± 5 seconds ($U(a,b)$, $a = mean - 5$, $b = mean + 5$). Moreover, the initial battery status does not necessarily have to be the same, so it is possible to model also, and we have set for this example a random offset (random value between *[0 mean]*) that has been included to model this phenomenon.

More information on these type of scenario is detailed in our previous work [46], [47], [49].

The source and destination nodes are assumed to be on the ground, and their battery will not be consumed during the experiment so that there is a continuous connection established between the source and the destination, and the routing protocol can be evaluated. For simplicity, the replacement period for each SUAV (i.e., the time needed for a UAV in the GCS to provide the service of the replaced UAV in the scenario) is fixed to 10 seconds (i.e., the mission planner usually coordinates UAV replacements), although VENUE allows any pattern to be included, and this value may vary for instance depending on the followed strategy from the control station or the size of the service provider's fleet, i.e., the number of available UAVs in the system.

The routing protocol used in this experiment as an example is Optimized Link State Routing Protocol (OLSR) [38]. Flying nodes store the updated list of destinations and the routes to them, and OLSR provides mechanisms to periodically refresh the routing tables and to maintain the network topology information.

An implementation of OLSR, Olsrd [50], has been selected to install the routing protocol in the Linux Containers. Olsrd is designed to be run as a standalone server process. All the configurable parameters are based on the standard values [38]. Each experiment has been repeated 30 times to obtain proper results.

C. METRICS

To evaluate the routing protocol, we characterize three relevant metrics. The first metric is the (i) packet loss that allows obtaining an idea of the convergence time required to find a path between the Source and Destination nodes, i.e., the more packets lost, the more time needed to update network paths. Another analyzed metric is the (ii) OLSR control traffic. This metric determines the overhead generated because of the routing protocol signaling, which may be critical in limited bandwidth systems. The third metric is the (iii) percentage of use of each SUAV. This metric reveals the significance of each UAV (depending on its position in the network). This metric may assist to the mission planner to get an insight about which UAVs are relevant, and in which UAVs the battery consumption can be higher, for example. Also, this section presents some network parameters that are important to identify what type of multimedia services can be deployed over the multi-UAV network, such as the jitter, the end-to-end delay, and the maximum available bandwidth. All these parameters can be easily obtained from the network traces generated by ns-3 and the network tool that can be integrated into VENUE (iPerf in this experiment).

D. RESULTS AND CONCLUSIONS

Before the previous metrics measurements are presented, and in order to better understand the results, Figure 3 shows a snapshot of the tests where it is possible to appreciate for a single experiment the moment in which the replacements of the SUAVs take place (bottom), and the corresponding received throughput at the destination SUAV (up). When an SUAV in the source-destination path is replaced, a consequent drop in the received throughput at the destination occurs. The figure reveals that the impact of the substitutions in the throughput depends on the replaced SUAV. When the replaced one is SUAV 4 (which connects the two parts of the network), the drop remains at least for 10 seconds, i.e., the replacement time. However, with the replacement of other SUAVs, the drop may remain for a smaller period (or not), depending on the time it takes the routing protocol to find another path. In a simple scenario such as the one proposed, it is not complex to identify which SUAVs have the most significant impact on the network service. However, in populated systems, this analysis helps to detect which are the SUAVs with an essential role in the FANET.

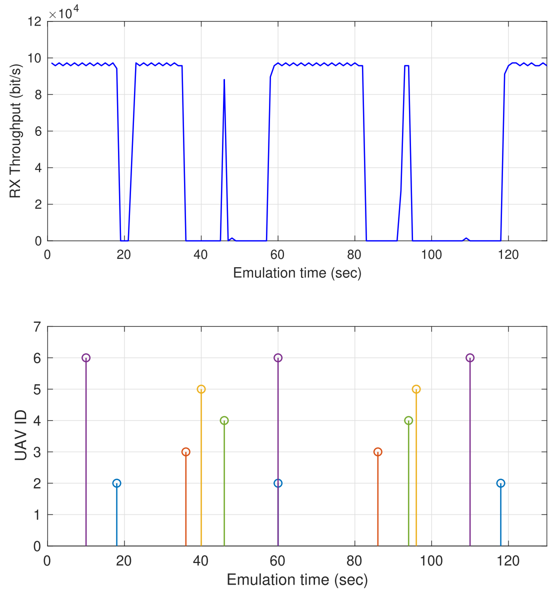


FIGURE 3. Received throughput at destination UAV in one of the experiments (fails each 50 seconds).

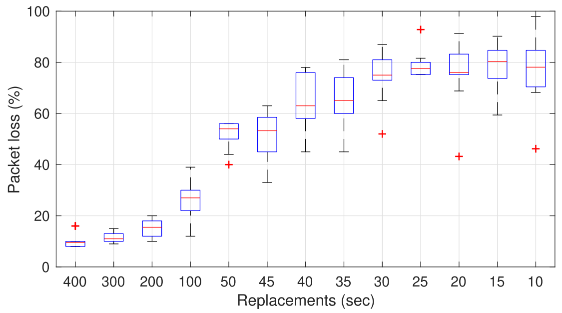
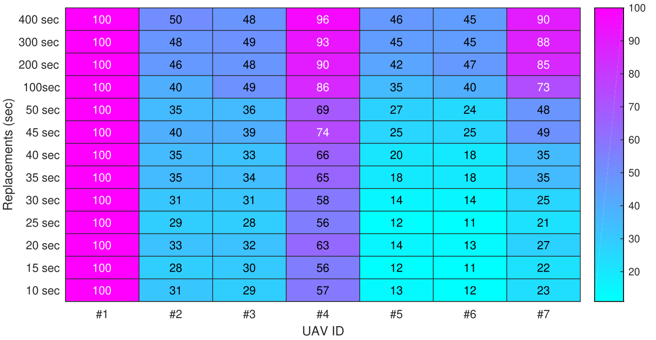


FIGURE 4. Packet loss percentage in the first scenario for different replacements values.

Figure 4 illustrates the percentage of packets lost in the system depending on the battery lifetime. To represent this data (corresponding to the 30 experiments), we have selected the boxplots. Through the boxplot, the center and dispersion of the data distribution can be easily perceived. The red line matches the median value of the data. The blue box (interquartile range) represents 50% of the data (from 75% to 25% of the values). Finally, the whiskers correspond to the rest of the values of the group. As expected, the more frequent the replacement is, the higher the packet loss percentage remains. Significantly, the value of the packet loss is always over 50%, with replacement times below 50 seconds. The replacements are indeed taken to the extreme since the UAVs battery-lifetime is way much longer in standard conditions. However, it can be concluded that for environments with severe changes, i.e., either due to very high mobility or failures in the links, the OLSR protocol may not perform correctly, while it performs quite acceptable with replacements over 100 seconds. This examination may be useful for scenarios where different routing protocols are considered assisting the mission planner in selecting the most suitable option to carry out the service efficiently.



Replacements (sec)	#1	#2	#3	#4	#5	#6	#7
400 sec	100	50	48	96	46	45	90
300 sec	100	48	49	93	45	45	88
200 sec	100	46	48	90	42	47	85
100sec	100	40	49	86	35	40	73
50 sec	100	35	36	69	27	24	48
45 sec	100	40	39	74	25	25	49
40 sec	100	35	33	66	20	18	35
35 sec	100	35	34	65	18	18	35
30 sec	100	31	31	58	14	14	25
25 sec	100	29	28	56	12	11	21
20 sec	100	33	32	63	14	13	27
15 sec	100	28	30	56	12	11	22
10 sec	100	31	29	57	13	12	23

UAV ID

FIGURE 5. Percentage of UAV Wi-Fi utilization.

Finally, Figure 5 presents the percentage of utilization of all the SUAVs defined as the data packets that pass through an SUAV divided by the total number of packets generated by the source. In the graph, it can be appreciated that SUAV 1 has 100% of utilization since it is the source that generates the traffic flow. It can also be seen that SUAV 4 has more usage than the rest of SUAVs because it connects the two parts of the multi-UAV network, and all the packets (except for the ones that are lost) go through it. This information, in combination with the knowledge obtained from Figure 3, is essential for the mission planner to be able to detect if a particular SUAV has a prominent role in the network and also to check if the system has been correctly designed.

TABLE 1. Aditional metrics.

Metrics		
Jitter	**Delay**	**OLSR Traffic**
3 ms	24 ms	5 Kbps

In Table 1 is represented other values that can help to determine what type of services can be deployed over the FANET network such as the average jitter (3 ms in our test scenario), the average delay from the source node to the destination node (24 ms), or the average control traffic throughput, i.e., OLSR routing protocol traffic sent by each node (5 Kbit/s).

V. USE CASE II: FANET TO EXTEND 5G CONNECTIVITY

The second use case is an example of how VENUE can serve to validate a much more complicated scenario: an NFV platform that uses a FANET to allow a smooth deployment of a VoIP service (including, for instance, the automatic deployment of virtualized SIP servers, DNS servers, or OLSR routers using OpenStack). Also, the VNFs have been instantiated into real hardware that is ready to be included as SUAV payload (Raspberry Pi). It must be clarified that a MANO system was not used to configure the VNFs in order to simplify the experiment explanation. However, detailed information about this configuration can be found in our previous work [46], [47].

In this experiment, VENUE will be used to emulate a scenario closer to the real flight conditions. In this scenario, it is not only expected that the VNFs can interact with each other or that the whole service or the FANET itself can be adequately established. All these things are assumed to have been tested in advance with regular trials. The main goal is to verify the scenario under the changing conditions that there will be when SUAVs take off. However, it will be possible to appreciate in the FANET the effects of the SUAVs movement, the loss of network connectivity implications in the routing protocols. It is particularly relevant (because these previous things can already be done with some existing simulators) to see the effect of this intermittent connectivity in the real deployed VNFs in the real services under execution and in the NFV orchestrator that is managing the whole service.

A. SCENARIO MOTIVATION

Nowadays, some of the new technologies that the 5G networks are promoting are the softwarization alternatives like the Network Function Virtualization (NFV) that allows to quickly deploy different network services on different devices, automatically orchestrating the distribution of virtual entities (Virtual Network Functions, VNF) through the network on the proper hardware.

Multi-UAV systems are being proposed as candidate alternatives to serve as computation nodes for these VNFs due to their flexibility to physically deploy the services wherever they are required: crowded events (e.g., concerts, demonstrations) where the conventional base station is overloaded or is not powerful enough, in emergencies, where connectivity is critical to ease emergency tasks or the physical infrastructure, does not exist, to support city areas with malfunctioning base stations, in search and rescue operations in remote areas (mountain, sea), etc. However, resource constraints that are inherent to SUAVs or their payloads (small single-board computers) have, as a consequence, the appearance of new challenges (for FANET to provide a stable and reliable service) that have to be solved before their correct deployment and operation. For this deployment, we used the prototype NFV system developed in our previous work [46], [47].

B. SCENARIO DESCRIPTION

This scenario includes a three SUAVs fleet and a ground control station intended to be used, for instance, to enable communications in emergencies as it can be seen in Figure 6. The service has been instantiated using a set of virtual functions and virtual networks that operate on top of the FANET to provide a flexible and dynamic connectivity backbone and service deployment.

Each SUAV will carry as payload a Raspberry Pi 3B (RPi), [51] single-board computer (SBC). All the RPis include an external battery-power supply (3.7 V and 3,800 mAh) so that the network service operation does not affect the SUAV battery itself (which is intensively required by the SUAV engines). The selected hardware is not a random choice. The small size of the RPi (85.60 mm×56 mm×21 mm), in combination with its reduced weight, allows almost any commercial UAV, e.g., DJI Phantom 4, to fly loading these devices without any problem. In Figure 7(b), it can be seen how the payload has been incorporated into the aircraft. In Figure 7(c), it can be appreciated the UAV flying with the payload. The ground control station is a mini-ITX computer (Intel Core i7 2.3 GHz, 16GB RAM, 128GB SSD, 4 GbE ports) that also acts as a cloud operating system (OpenStack [52]). This equipment can be appreciated in Figure 7(a). As it is shown in Figure 6, the physical network topology is emulated using VENUE (VENUE emulates the realistic conditions of the wireless ecosystem), and on top of the network, a virtual network service has been deployed. Two SUAVs (1 and 2) also provide real Wi-Fi access points enabling end-users to utilize the network. The created (emulated by VENUE) Wi-Fi network follows the 802.11a standard. The created (real) access points follow the 802.11n standard. VENUE provides mobility to

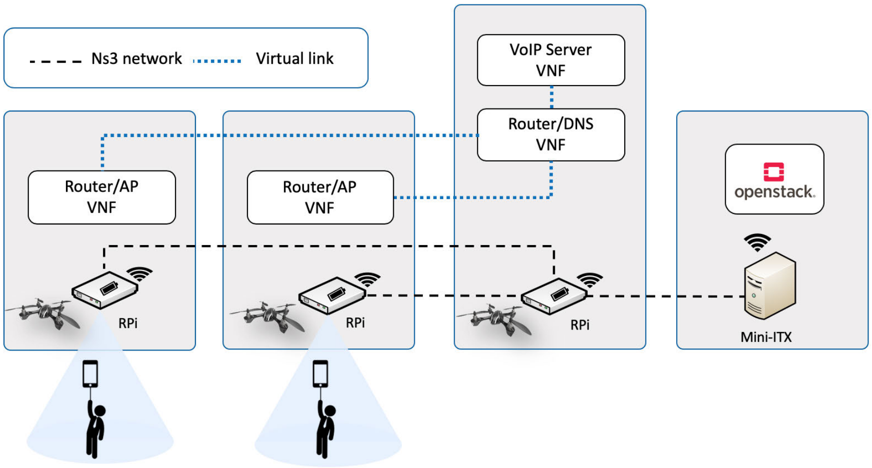
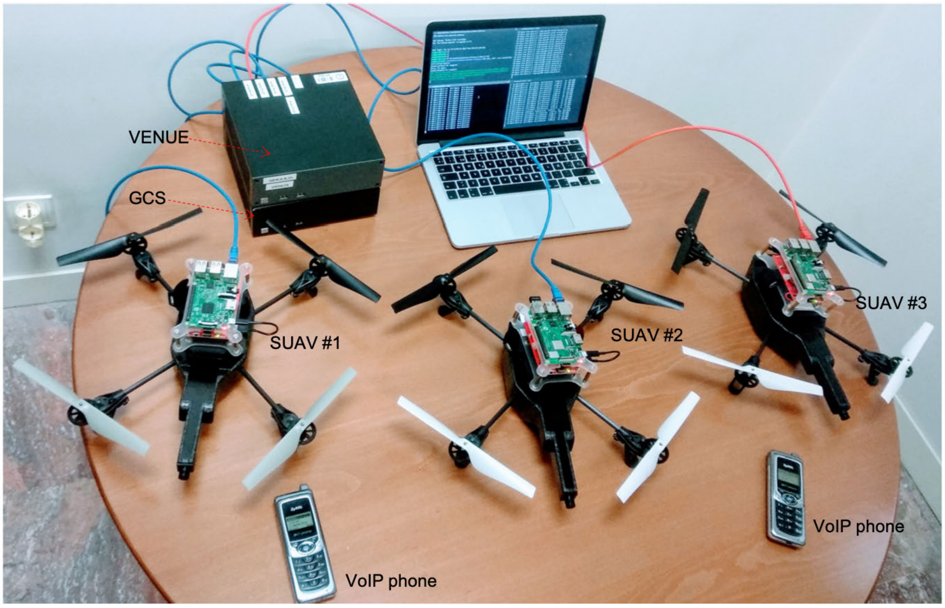


FIGURE 6. Scenario II: multi-UAV network to extend 5G connectivity.



(a) Scenario II Testbed: SUAVs, Single-board Computers, GCS, VoIP terminals and VENUE emulation platform

(b) DJI Phantom 4 with RPI and battery as payload

(c) DJI Phantom 4 flying with RPI and battery as payload

FIGURE 7. Real hardware used in the experiments.

the commodity equipment, as can be appreciated in the following section figures. Otherwise, making these scheduled replacements would not be possible.

In order to create a network service, different VNFs have been used: (i) Two VNFs implement the router functionality, (ii) another VNF implements a router and also includes a DNS service and finally, (iii) another one implements a Voice-over-IP server based on the Session Initiation Protocol (SIP) [53] server (Kamailio [54]), to allow the ground users to ‘‘register’’ wireless terminals in the VoIP server and maintain telephone conversations with other users. On the other hand, two APs have been configured (including a DHCP server) that allow users to connect to the network deployed by the SUAVs.

All these VNFs are instantiated and configured using OpenStack. The ground control station and the devices hosting the deployed VNFs use OLSR to enable the communications. Remarkably, the virtual networks are deployed using the Virtual eXtensible Local Area Networks (VXLAN) [55].

A complete VoIP call (including the signaling process to start the call) has been performed to test the whole network service. The ZyXEL Prestige 2000W terminals have been utilized to make the call. Besides, a video stream is also sent through the network using the VLC [56] tool. Finally, during the mission, SUAV 2 is replaced to force an additional topology change.

C. RESULTS AND CONCLUSIONS

Network traffic is captured using the Wireshark tool to analyze this scenario. Captures are performed in SUAV 3 both in the ad hoc network interface (emulated network by the VENUE platform) and in the AP interface (real), in order to analyze all the traffic generated by the service.

Figure 8 shows the control traffic between the OpenStack controller and SUAV 3 compute node necessary to manage the computing, storage, and networking resources of the NFV platform. As a result of the replacement of SUAV 2, the figure reveals a drop in the received traffic while the computer node keeps on sending traffic to the controller steadily.

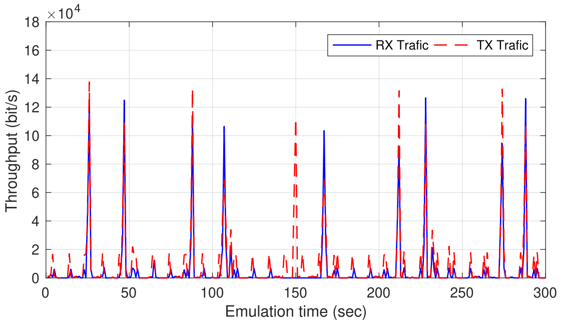


FIGURE 8. OpenStack control traffic in a compute node.

Similarly, OLSR signaling traffic can also be appreciated in Figure 9. Likewise, during SUAV 2 replacement, there is a stop in the received traffic. OLSR average throughput is around 3 Kbit/s (which is negligible as compared to multimedia services).

Figure 10 illustrates SIP signaling traffic to start the multimedia VoIP call. Moreover, the DNS traffic is also represented in the figure. The DNS is required to resolve the name of the SIP server and register the VoIP terminals.

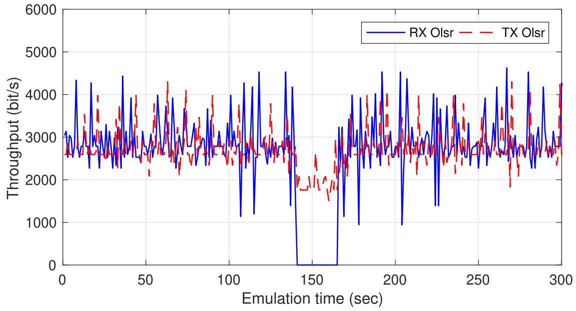


FIGURE 9. OLSR signaling traffic at the compute node.

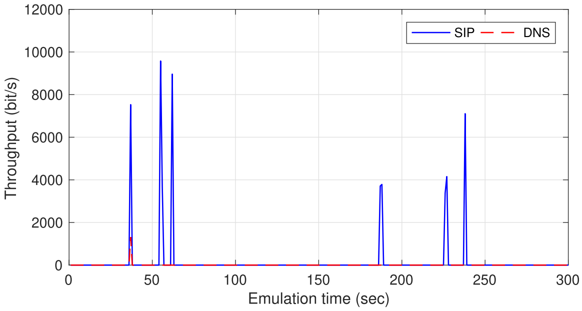


FIGURE 10. SIP and DNS traffic needed for the VoIP call.

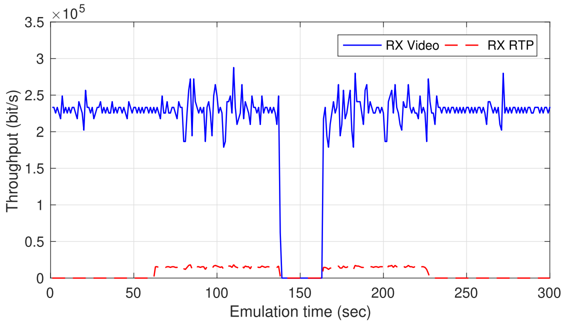


FIGURE 11. Received video and audio at the destination.

Finally, Figure 11 reflects the voice traffic received by one of the wireless phones. The call was made without errors and with appropriate sound quality. Figure 11 also shows the received video traffic at the destination. As can be seen, after SUAV 2 replacement, the two services are correctly recovered.

VI. CONCLUSION AND FUTURE WORK

This article has presented VENUE: an emulation platform for FANETs that enables the validation of different network service deployments in multi-UAV systems. VENUE provides a controllable and reproducible environment that allows experimenters to extract multiple conclusive and reliable results.

VENUE is based on the ns-3 simulation software and Linux Containers lightweight VNFs. Furthermore, the platform allows the VNFs to interact with general-purpose UAV equipment. However, for this aim, the source code of ns-3 needs to be modified (a patch file including the corrections is also provided in [33]). This functionality allows several benefits, such as to add realistic energy consumption models based on measurements or the integration of the simulated network with a 5G core.

As a differentiation factor in comparison to similar developments, VENUE framework includes the ns-3 simulator as a VNF, which enables modular prototyping; it also incorporates pre-created Linux Containers already configured with FANET routing protocols (OLSR and SDN) to promote the use of the platform. Finally, mobility has been incorporated into the network nodes to simulate a plausibility multi-device environment.

To conclude, the article validates VENUE functionalities in two different use cases. The first use case verifies the integration of real software (virtual entities in this case) with VENUE platform and evaluates FANET routing protocols. The second use case validates a network service functionality deployed on top of real UAV hardware. The results shown in the proposed use cases reveal that VENUE platform is suitable for the prototyping and development of multi-UAV systems, reducing to the gap between the development and production stage.

There are some developments that are expected to be included in VENUE in the near future. The first one is to provide different mobility models to cover multiple UAV applications. By including this functionality, the user of the platform does not need to precompute any mobility pattern, boosting the platform usage. Second, to provide LXC with the preinstallation of diverse routing alternatives, covering from the most popular MANET protocols, e.g., OLSR, AODV, DSDV, BATMAN, to other innovative solutions applied, particularly to UAVs. Finally, the intention is to include the development of a configurable SDN based extension to allow testing different alternatives to face the routing challenge in FANETs that also takes into account the efficient energy consumption. As soon as multi-UAV services begin to be widely deployed, flexible FANET modeling platforms like VENUE will be increasingly useful to facilitate all these deployments.

REFERENCES

[1] *FAA Aerosp. Forecasts*. Accessed: Sep. 19, 2019. [Online]. Available: https://www.faa.gov/data_research/aviation/aerospace_forecasts/

[2] *CORUS*. Accessed: Sep. 19, 2019. [Online]. Available: https://www.eurocontrol.int/project/concept-operations-european-utm-systems

[3] *Unmanned Aircraft System Traffic Management (UTM)*. Accessed: Sep. 19, 2019. [Online]. Available: https://www.faa.gov/uas/research_development/traffic_management/

[4] S. Long. *Drones and Precision Agriculture: The Future of Farming*. Accessed Sep. 19, 2019. [Online]. Available: https://www.microdrones.com/en/content/drones-and-precision-agriculture-the-future-of-farming/

[5] N. Athanasis, M. Themistocleous, K. Kalabokidis, and C. Chatzitheodorou, "Big data analysis in UAV surveillance for wildfire prevention and management," in *Proc. Eur., Medit., Middle Eastern Conf. Inf. Syst.* Cham, Switzerland: Springer, Oct. 2018, pp. 47–58.

[6] *Professional Society of Drone Journalists*. Accessed: Sep. 19, 2019. [Online]. Available: http://www.dronejournalism.org/

[7] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.

[8] A. Fouda, A. S. Ibrahim, I. Guvenc, and M. Ghosh, "UAV-based in-band integrated access and backhaul for 5G communications," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.

[9] H. Niu, N. Gonzalez-Prelcic, and R. W. Heath, Jr., "A UAV-based traffic monitoring system—Invited paper," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.

[10] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, "UAV-assisted supporting services connectivity in urban VANETs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3944–3951, Apr. 2019.

[11] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[12] A. Mukherjee, V. Keshary, K. Pandya, N. Dey, and S. C. Satapathy, "Flying ad hoc networks: A comprehensive survey," in *Information and Decision Sciences*. Singapore: Springer, 2018, pp. 569–580.

[13] B. Barritt, T. Kichkaylo, K. Mandke, A. Zalcman, and V. Lin, "Operating a UAV mesh & Internet backhaul network using temporospatial SDN," in *Proc. IEEE Aerosp. Conf.*, Mar. 2017, pp. 1–7.

[14] K. J. White, E. Denney, M. D. Knudson, A. K. Mamerides, and D. P. Pezaros, "A programmable SDN+NFV-based architecture for UAV telemetry monitoring," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 522–527.

[15] M. Sara, I. Jawhar, and M. Nader, "A softwarization architecture for UAVs and WSNS as part of the cloud environment," in *Proc. IEEE Int. Conf. Cloud Eng. Workshop (ICEW)*, Apr. 2016, pp. 13–18.

[16] *Network Functions Virtualization (NFV); Architectural Framework V1.1.1*, document ETSI GS NFV 002, ETSI, Oct. 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf

[17] *HOWTO Use Linux Containers to Set Up Virtual Networks*. Accessed Sep. 19, 2019. [Online]. Available: https://www.nsnam.org/wiki/HOWTO_Use_Linux_Containers_to_set_up_virtual_networks

[18] *The Network Simulator—NS-2*. Accessed: Sep. 19, 2019. [Online]. Available: https://www.isi.edu/nsnam/ns/

[19] G. F. Riley and T. R. Henderson, "The $NS-3$ network simulator," in *Modeling and Tools for Network Simulation*. Berlin, Germany: Springer, 2010, pp. 15–34.

[20] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simulation Tools Techn. Commun., Netw. Syst. Workshops*, Mar. 208, Art. no. 60.

[21] *Mininet: An Instant Virtual Network on Your Laptop (or Other PC)-Mininet*. Accessed: Sep. 19, 2019. [Online]. Available: http://mininet.org/

[22] M. Pužar and T. Plagemann, "NEMAN: A network emulator for mobile ad-hoc networks," in *Proc. 8th Int. Conf. Telecommun. (ConTEL)*, Zagreb, Croatia, 2005, pp. 155–161. doi: 10.1109/CONTEL.2005.185841.

[23] M. A. To, M. Cano, and P. Biba, "DOCKEMU —-A network emulation tool," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2015, pp. 593–598.

[24] J. Zhang and Z. Qin, "TapRouter: An emulating framework to run real applications on simulated mobile ad hoc network," in *Proc. 44th Annu. Simulation Symp.*, Apr. 2011, pp. 39–46.

[25] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—A modular gazebo MAV simulator framework," in *Robot Operating System (ROS)*. Cham, Switzerland: Springer, 2016, pp. 595–625.

[26] N. P. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Sep. 2004, pp. 2149–2154.

[27] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, "OpenUAV: A UAV testbed for the CPS and robotics community," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 130–139.

[28] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 445–461.

[29] K. Singh and A. K. Verma, "Experimental analysis of AODV, DSDV and OLSR routing protocol for flying adhoc networks (FANETs)," in *Proc. IEEE Int. Conf. Electr., Comput. Commun. Technol. (ICECCT)*, Mar. 2015, pp. 1–4.

[30] K. Singh and A. K. Verma, "Applying OLSR routing in FANETs," in *Proc. IEEE Int. Conf. Adv. Commun., Control Comput. Technol.*, May 2014, pp. 1212–1215.

[31] A. Nayyar, "Flying adhoc network (FANETs): Simulation based performance comparison of routing protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP," in *Proc. Int. Conf. Adv. Big Data, Comput. Data Commun. Syst. (icABCD)*, Aug. 2018, pp. 1–9.

[32] H. Nawaz, H. M. Ali, and S. U. R. Massan, "A study of mobility models for UAV communication networks," in *Proc. 3C Tecnologia*. Rochester, MN, USA: Mayo, May 2019, pp. 276–297.

[33] V. Sanchez-Aguero. *VENUE*, Accessed: Sep. 19, 2019. [Online]. Available: https://github.com/vsaguero/VENUE

[34] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81057–81105, 2019.

[35] M. O. Kalinin, V. Krundyshev, and P. Semianov, "Architectures for building secure vehicular networks based on SDN technology," *Autom. Control Comput. Sci.*, vol. 51, no. 8, pp. 907–914, Dec. 2017.

[36] R. Kumar, M. A. Sayeed, V. Sharma, and I. You, "An SDN-based secure mobility model for UAV-ground communications," in *Proc. Int. Symp. Mobile Internet Secur.* Springer, 2017, pp. 169–179.

[37] Z. Kaleem and M. H. Rehmani, "Amateur drone monitoring: State-of-the-art architectures, key enabling technologies, and future research directions," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 150–159, Apr. 2018.

[38] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol 1012 (OLSR)*, vol. 51, document RFC3626, 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[39] *Ryu SDN Framework*. Accessed: Sep. 19, 2019. [Online]. Available: https://osrg.github.io/ryu/

[40] *Open vSwitch*. Accessed: Sep. 19, 2019. [Online]. Available: https://www.openvswitch.org/

[41] GUEANT, *IPERF—The TCP, UDP and SCTP Network Bandwidth 1018 Measurement Tool*. Accessed: Sep. 19, 2019. [Online]. Available: https://osrg.github.io/ryu/

[42] *TCPDUMP/LIBPCAP Public Repository*. Accessed: Sep. 19, 2019. [Online]. Available: https://www.tcpdump.org/

[43] U. Lamping and E. Warnicke, "Wireshark user's guide," in *Interface*, vol. 4, no. 6. 2004.

[44] *Linux Containers*. Accessed: Sep. 19, 2019. [Online]. Available: https://linuxcontainers.org/

[45] *Linux Containers—LXD—Introduction*. Accessed: Sep. 19, 2019. [Online]. Available: https://linuxcontainers.org/lxd/

[46] B. Nogales, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso, "A NFV system to support configurable and automated multi-uav service deployments," in *Proc. 4th ACM Workshop Micro Aerial Vehicle Netw., Syst., Appl.*, Jun. 2018, pp. 39–44.

[47] B. Nogales, V. Sanchez-Aguero, I. Vidal, and F. Valera, "Adaptable and automated small UAV deployments via virtualization," *Sensors*, vol. 18, no. 12, p. 4116, Nov. 2018.

[48] *Tap Net Device âĂŤ NS-3 NS-3.14 Documentation*. Accessed Sep. 19, 2019. [Online]. Available: https://www.nsnam.org/docs/release/3.14/models/html/tap.html

[49] C. Tipantuña, X. Hesselbach, V. Sánchez-Aguero, F. Valera, I. Vidal, and B. Nogales, "An NFV-based energy scheduling algorithm for a 5G enabled fleet of programmable unmanned aerial vehicles," *Wireless Commun. Mobile Comput.*, vol. 2019, Feb. 2019, Art. no. 4734821. doi: 10.1155/2019/4734821.

[50] *OLSR.org Wiki*. Accessed: Sep. 19, 2019. [Online]. Available: http://www.olsr.org/mediawiki/index.php/Main_Page

[51] *RPI*. Accessed Sep. 19, 2019. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[52] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "OpenStack: Toward an open-source solution for cloud computing," *Int. J. Comput. Appl.*, vol. 55, no. 3, pp. 38–42, 2012.

[53] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *SIP: Session Initiation Protocol*, document RFC 3261, 2002.

[54] *Kamailio SIP Server*. Accessed Sep. 19, 2019. [Online]. Available: https://www.kamailio.org/w/

[55] M. Mahalingam, D. G. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, *Virtual Extensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks Over Layer 3 Networks*, document RFC 7348, 2014, pp. 1–22.

[56] *Offcial Download of VLC Media Player, the Best Open Source Player 1059- VideoLAN*. Accessed Sep. 19, 2019. [Online]. Available: https://www.videolan.org/vlc/index.es.html

VICTOR SANCHEZ-AGUERO received the B.Sc. degree in audiovisual systems engineering and the M.Sc. degree in telematics engineering from the University Carlos III of Madrid (UC3M), in 2017 and 2018, respectively. He is currently pursuing the Ph.D. degree with the IMDEA Networks Institute. He was an Internship Student with the Saïd University collaborating with researchers from the Telematics Engineering Department. He has published different articles in his research field in different national and international conferences and journals.

FRANCISCO VALERA received the Telecommunication Engineering degree from the Technical University of Madrid (UPM), in 1998, and the Ph.D. degree in telecommunications from the University Carlos III de Madrid (UC3M), in 2002. He is currently a Tenured Associate Professor and the Head of the Telematics Engineering Department with UC3M. He has published over 80 articles in the field of advanced communications in magazines and conferences.

BORJA NOGALES received the bachelor's degree in telecommunication technologies engineering from the University Carlos III de Madrid (UC3M), in 2016, where he is currently pursuing the Ph.D. degree in telematics engineering. He is involved in the European research project 5GinFIRE and the national project 5GCity. His research interests include network functions virtualization (NFV), 5G networking, and unmanned aerial vehicles (UAVs).

LUIS F. GONZALEZ is currently pursuing the Ph.D. degree in telematics engineering with the University Carlos III de Madrid. He is involved in the European research project 5GinFIRE and the national project 5GCity. His research interests include network functions virtualization (NFV), 5G networking, and unmanned aerial vehicles (UAVs).

IVAN VIDAL received the Ph.D. degree in telematics engineering from the University Carlos III de Madrid, in 2008. He is currently a Visiting Professor with the University Carlos III de Madrid. His research interests include unmanned aerial vehicles (UAVs), 5G networks, and multimedia networking. He has been involved in several international and national research projects, including the H2020 5GinFIRE and 5GCity. He has published more than 50 scientific articles in several conferences and international journals.