González, Luis F., et al. NFV orchestration on intermittently available SUAV platforms: challenges and hurdles, *in IEEE INFOCOM 2019 Workshops - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Pp. 301-306*
DOI: https://doi.org/10.1109/INFCOMW.2019.8845040

NFV orchestration on intermittently available SUAV platforms: challenges and hurdles

Luis F. Gonzalez[1], Ivan Vidal[1], Francisco Valera[1], Victor Sanchez-Aguero[1,2], Borja Nogales[1], and Diego R. Lopez[3].

[1] University Carlos III of Madrid, Av. Universidad, 30, 28911, Leganés (Madrid), Spain
[2] IMDEA Networks Institute. Avda. del Mar Mediterráneo, 22, 28918, Madrid, Spain
[3] *Telefónica I+D, C/ Zurbarán 12, 28010 Madrid, Spain*
Email: luisfgon@it.uc3m.es; ividal@it.uc3m.es; fvalera@it.uc3m.es; victor.sanchez@imdea.org; bdorado@pa.uc3m.es; diego.r.lopez@telefonica.com.

Abstract— **In this paper, we analyze the main challenges and issues related with the orchestration of Virtualized Network Functions (VNFs) on Small Unmanned Aerial Vehicles (SUAVs). Our analysis considers a reference scenario where a number of SUAVs are deployed over a delimited geographic area and provide a mobile cloud environment that supports the deployment of functionalities using Network Functions Virtualization (NFV) technologies. The orchestration of services in this reference scenario presents different challenges, due to the constrained capacity and limited lifetime of battery-powered SUAVs, the intermittent availability of network communications, and the need to consider enhanced policies for the allocation of virtual functions to SUAVs. Finally, we perform a first exploratory evaluation of the identified challenges and issues, using a well-known and widely adopted virtualized infrastructure manager, i.e., OpenStack.**

Keywords— NFV, Management and Orchestration (MANO), SUAV, intermittent availability.

I. INTRODUCTION

It is expected that the next generation of mobile networks, or 5G, will soon revolutionize the world of telecommunications, introducing significant benefits in comparison with the previous mobile network standards, particularly focusing on increased data rates to support the data demand, improved latency of wireless communications, and reduced costs in terms equipment and operations [1]. Network Functions Virtualization (NFV) is one of the main enablers to achieve these objectives, providing both hardware standardization and service softwarization to reduce deployment and maintenance costs, as well as easing the development of network services.

On the other hand, Small Unmanned Aerial Vehicles (SUAVs) are nowadays proliferating in the market, enabling a wide set of novel and appealing civilian and military applications, such as those present in [2] like providing a mobile infrastructure in disaster scenarios or a surveillance system using SUAVs with cameras. Given their capacity to onboard computing, storage and networking devices, they have recently started to obtain significant attention from the research community, as a platform to flexibly support cost-effective data communications in the scope of 5G networks.

However, despite the many efforts that have been put in developing resource orchestration functionalities in the 5th generation of mobile networks, there still seems to be important challenges and hurdles that need to be addressed to effectively support NFV in wireless Ad-hoc network scenarios, such as those that may be enabled by SUAV platforms. Motivated by this observation, this paper aims at identifying and analyze these challenges, and anticipate some possible directions to address them

The rest of this paper is structured as follows: in section II, we review the main concepts behind NFV orchestration, and we review the existing literature regarding both orchestration and SUAVs. In section III, we do a theoretical analysis of all problems and bottlenecks that current orchestration might face while presenting some alternatives for its solution, separating this section into different part to discuss all issues separately. In section IV, we emulated a Flying Ad-hoc Network (FANET) scenario using a well-known orchestrator to test the how a well-known solution deals with intermittently available SUAVs platforms and analyze the results obtained. In section V, we present the main conclusions withdrawn from our work, as well as a discussion to continue developing NFV orchestration on intermittently available SUAVs platforms in the future.

II. BACKGROUND AND RELATED WORK

Given the relevance of NFV as a key and enabling technology to support the softwarization of network functions and components, the European Telecommunications Standard Institute (ETSI) created the NFV Industry Specification Group, to provide a reference architectural framework for NFV deployments [3]. Virtualized Network Functions (VNFs) are central elements of this architecture. A VNF is a software implementation that provides the functionality of a network function (e.g., a router, as voice-over-IP server, or a load balancer). VNFs are deployed over an NFV infrastructure (NFVI), which provides the hardware and software resources that are needed to create an appropriate virtual environment to support their execution. The NFVI provides an abstraction layer that enables the separation of the VNF's functionality from the used hardware. This avoids the necessity of having specialized equipment for every type of service, saving costs and simplifying both the development of network functions and their deployment. VNFs are interconnected to effectively build up end-to-end Network Services (NSs), easing the process of provisioning service chains and applications by telecommunication operators and Service Providers.

To coordinate the operations over the NFV environment, the ETSI NFV reference architecture defined a Management and Orchestration (MANO) system. This is in turn divided into three main components: 1) the Virtualized Infrastructure Manager (VIM), which controls and manages the resources of an NFVI; 2) the Virtual Network Function Manager (VNFM), responsible of the instantiation of VNFs, as well as the configuration, modification and termination of VNF instances; and 3) the NFV Orchestrator (NFVO), which orchestrates the allocation of resources under the control of different VIMs, and manages the lifecycle of network services [4].

SUAVs have started to gain attention as an enabling platform to support data communications in the scope of 5G networks. In this respect, the work in [5] explores the utilization of SUAVs as 5G points of presence, with the capacity to onboard computing, storage and networking resources that support a cost-effective deployment of network infrastructure over delimited geographic areas. This concept was further elaborated in [2] and [6], where the authors present the design of an NFV system capable of supporting the agile configuration and deployment of moderately complex network services over a cloud platform offered by a swarm of resource constrained SUAV equipment.

The potential of NFV has recently received attention from the research community to support the flexible deployment of applications and functions over UAV platforms. For example, authors in [7] exploit UAV's mobility to provide a full video-surveillance system in big poorly internet-covered areas using NFV, transmitting video signal through a UAV network with VNFs deployed in the aircrafts. They propose implementing its behavior using paravirtualization, where Virtual Machines (VMs) share the hardware directly, allowing its host Operative System to only be a platform for operating with the VNFM. Moreover, some research has been focused on providing seamless transition between UAVs in migration cases, which can only be achieved if all associated network services, routing and operational control migrate rapidly as well. Authors in [8] propose an NFV-based solution that also takes into account the high-mobility requirements of these networks.

III. Problem Statement and Analysis

While the previous works represent significant steps towards supporting the softwartization of functions over SUAV platforms, the realization of such a view, where SUAVs provide a programmable NFV infrastructure that enables the automated deployment and the operation of NSs, still requires a careful analysis of diverse and challenging aspects. This section identifies these challenges and potential issues while pointing out research directions that could be followed to address them.

In our analysis, we assume a reference scenario where a number of SUAV units are deployed over a delimited geographic area. Each of these SUAVs provides a set of computing, storage and network resources, which conform an NFVI under the control of a VIM (i.e., each SUAV is an NFVI node). SUAVs can be interconnected using wireless technologies (e.g., WiFi or line-of-sight radio links), building a FANET that enables multi-hop data communications over the geographic area (e.g., real-time audio communications between users in the area). SUAVs may be placed at static positions or move, either autonomously or instructed by an operator from a Ground Control Station (GCS) (the movement of SUAVs can be for instance necessary in search and rescue operations, or in road traffic monitoring [9]). An NFV orchestrator interacts with the VIM and coordinates the automatic deployment of NSs over the NFVI conformed by the SUAVs. This way, SUAVs provide an adaptable platform that can be used in different use cases. Given their criticality, the NFV orchestrator and the VIM are hosted at the GCS. An example of this architecture can be seen in Fig. 1.

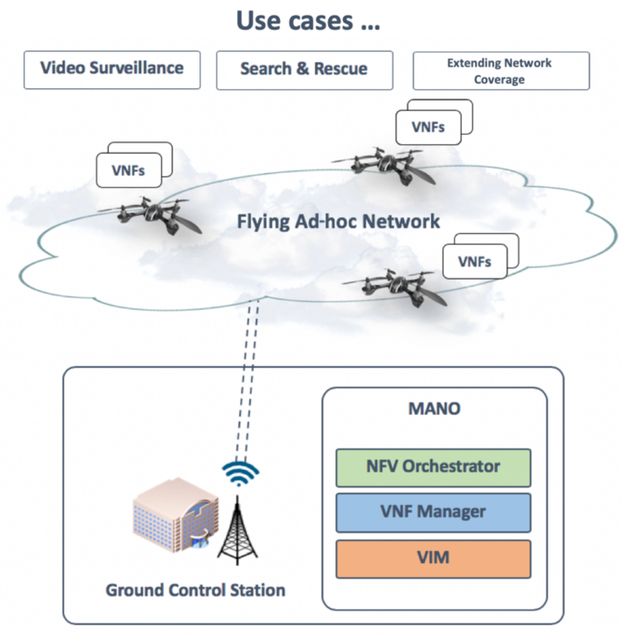

Fig. 1. SUAV network use cases.

A. *Limited lyfetime of NFVI nodes*

The utilization of an NFVI composed of SUAVs presents diverse challenges that have to be considered for effectively orchestrating NFV services. One critical aspect to be taken into account is the battery life. When these devices are in the air they consume battery, even when they might not be executing any networking/processing task. Therefore, when battery is running out, the VNFs hosted by a SUAV need to be migrated to another SUAV with sufficient energy capacity. In those cases where VNFs need to be placed at specific locations (e.g., a network router at a concrete GPS position), the migration of the VNFs may require the replacement of the SUAV unit by another one. This is challenging due to the limitations on the compute, storage and network resources that may be needed for the migration. In addition, the migration of the VNFs should be anticipated to guarantee a seamless replacement of the SUAV, or at least to minimize the disruption time caused by the unavailability of the affected VNFs.

Hence, in the considered scenario, the status of the battery at every SUAV becomes as relevant as the status of the compute, storage and network resources, and should be monitored by the VIM in order to estimate the remaining battery lifetime of the SUAVs and reduce the disruption time caused by the handover of the VNFs running on top of it to another SUAV. Additionally, and as an important consideration, the monitoring process carried out by the VIM to verify the status of the resources at the different SUAV units should not impose a significant increase of the battery consumption.

B. *Intermittent availability of control communications*

SUAVs battery consumption turns network nodes into non-permanent units, i.e., they become volatile nodes that can be replaced by other nodes. It is important to highlight that, in the considered scenario, where SUAVs form a multi-hop wireless Ad-hoc network, a node going offline may cause a disruption of the communications between other SUAVs and the ground control station (i.e., those communications that use the failed node as a relay). This disruption is likely to be transient, while the existing routing protocol implemented by the SUAVs converges and establishes new end-to-end network paths between the affected SUAVs and the VIM. However, during this period, non-reachable SUAVs are unavailable to the MANO system. This is a challenge to NFV orchestration, as the MANO system cannot deploy nor configure VNFs over those SUAVs, despite the involved NFVI resources being online but transiently unreachable. Hence, it is vital to ensure that the orchestrator is able to detect these malfunctions correctly and not interpret these failures as permanent, supporting a reasonable delay in the execution of the orchestration actions as along as this is permitted by the time restrictions of the use case. Otherwise NFV coordination of all network services could be sub-optimal at best and impossible at worse (if links never recover properly, no orchestration is possible).

It is important to highlight that temporary unavailability of NFVI nodes can be fairly common in the reference scenario under consideration. On the one hand, communication between UAVs is done through wireless media, a far less reliable medium compared to completely wired scenarios. Depending on network's placement and the medium itself, this can produce disconnections and reconnections of the wireless links. On the other hand, the mobility of SUAVs in certain use cases (e.g., in search and rescue operation) may introduce changes to the network topology of the aerial network, causing the temporary unavailability of NFVI nodes.

C. *Limited-capacity of NFVI nodes*

We must bear in mind that the utilization of SUAVs imposes restrictions on the size and weight of the equipment that can be onboarded as the aircraft payload. This inevitably introduces limitations on the computing, storage, and network resources that can be contributed by each SUAV to the NFVI, which may be limited to a set of single board computers. As discussed in [2], this might encourage the utilization of lightweight VNFs and container virtualization, as opposed to traditional hypervisor-based virtualization. A possible alternative to these containers could be using paravirtualization, where VM directly communicates with its hypervisor instead of communicating with its "virtualized kernel", speeding the interchange of information between hosts infrastructure and the VM/VNF. This way, the OS of the hosts can be used as a mere communication enabler between the VNFs and the VNFM, increasing orchestration efficiency. This idea is proposed by authors in [7].

On the other hand, how VIMs exchange information with each of the UAVs might have an impact on the overall node's performance. For the configuration and monitoring of virtual functions, most commercial VIMs use the HTTP protocol to send actions and/or request certain information between the VIM and nodes with VNFs. This may be problematic because HTTP was not designed as a lightweight protocol, aiming at operating on small devices with reduced computing power. As an alternative to HTTP, a less process-intensive protocol like CoAP [10] or MQTT [11] could be utilized instead. This would help reducing message length and optimizing their processing in constrained devices, providing a more cost-efficient solution in terms of battery consumption and communications overhead (the latter would be especially relevant in case of large deployments with multiple SUAVs).

D. *Tranport-layer protocols for control communications*

Focusing on the control communications that take place between the VIM and the SUAVs that conform the NFVI,

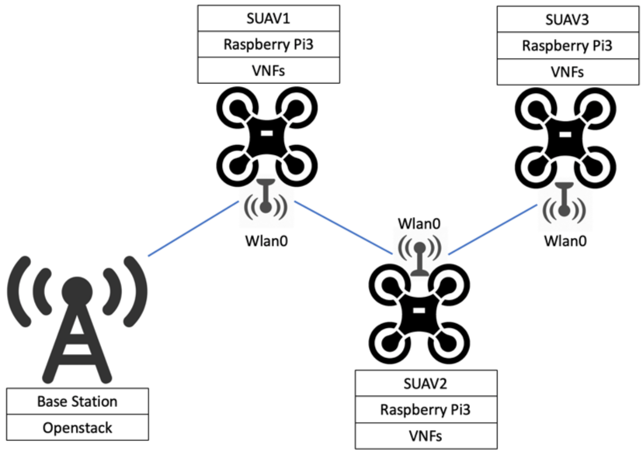

Fig. 2. Description of the experimental setup.

there are specific aspects related with HTTP that deserve special consideration. Concretely, HTTP uses TCP as its default transport-layer protocol. However, existing works that evaluate the performance that can be achieved by TCP over mobile Ad-hoc networks [12-16] show that TCP may not be an appropriate choice for these networks. On its conception, TCP was developed as a protocol for reliable data transfer over wired networks, where packet losses would mostly be related to congestion, and link-related failures would rarely occur. This changed with the entrance of wireless technologies, challenging TCP's core design with new application scenarios. A description of the issues related the utilization of TCP in wireless environments can be found in [12-13]. The most relevant for our analysis are:

- At the physical layer: Wireless links usually have a greater bit error rate compared to wired networks. Problems like fade-away and interferences can produce errors on packets, and poor-quality links induce more packet losses, severely harming its performance [12]. This has a noticeable impact on its behavior, as sender will always need to retransmit more packets as consequence of the amount of packet or/and acknowledgment losses, keeping sender's congestion window low and reducing overall throughput as a result [13].

- At the network layer: Ad-hoc networks require dynamic routing protocols like Dynamic Source Routing (DSR) [17] or Ad-hoc On-Demand Distance Vector (AODV) [18] to compute new routes when one or more routes are not available. As we have seen, this may occur due to the unavailability of nodes caused by battery exhaustion, or as a consequence of the mobility of SUAVs (authors in [14] studied how choosing a different mobility pattern can affect the performance of TCP). In some cases, TCP's retransmission timeout might be smaller than the time used for recomputing those routes, so sender will try to use an outdated route and retransmit one or more packets. When no acknowledgement is received, it will invoke its congestion control and reduce the congestion window in response. Even when the route is successfully recomputed again, and sender is able to send packets again, TCP's slow start mechanism will prevent the sender to have the same packet rate before route's failure, reducing its throughput. The presence of more intermediate nodes between sender and client further decreases performance, as intermediate nodes have to recompute those routes too and might experience the same problems. This problem is described in detail in [15].

All the aforementioned problems can be summarized in one main idea: TCP is unable to recognize the source of a failure in wireless media. In consequence, it treats this error as a congestion problem and invokes the corresponding control mechanism. In wired networks this is understandable, as errors related to failing links are usually scarce. In our scenario however, it may be problematic, as frequent node connectivity interruptions can decrease throughput if TCP reacts to link interruptions by using its congestion control mechanism.

So far, our discussion on the adequateness of TCP has mainly been focused on the control communications between the VIM and the SUAVs. However, the VIM is not the only MANO component that needs connectivity with the nodes of the NFVI: VNFM entities may also exchange control information with the VNFs. Hence, the implementation of a VNFM, operating under our reference scenario, should take into account the aforementioned considerations regarding the use of TCP.

Our discussion on the utilization of TCP in our reference scenario highlights a set of challenges we need to overcome to support reliable control communications between the MANO system and the NFVI nodes. We need to provide a reliable data service to ensure that orchestration actions and commands safely arrive to every node of the network and vice versa. However, this reliability should not come at the cost of harming the overall throughput and the performance of control communications. In this respect, a possible approach could be to use a datagram-oriented solution at the transport layer (i.e., UDP), and support the reliability of the data transfer at the application layer. That is, reliability would be implemented by the application processes running at the MANO system and the SUAVs, which would be tailored to the specifics of the considered use case. These applications could still use an application-layer protocol following a REST model like HTTP. CoAP is an example of such a protocol, sharing the REST model of HTTP, but operating over UDP and providing reliability at the application layer, making it a potential alternative to HTTP/TCP to support the exchange of control communications in our reference scenario.

E. Enhanced policies for the placement of VNFs

Aside from the aspects related to traffic exchange, there are additional challenges that need to be addressed to support effective orchestration actions in our reference scenario. Although VIMs take into account certain parameters to guide the allocation of virtual functions to NFVI nodes (e.g., CPU usage, memory, etc.), battery lifetime is not used to select the appropriate unit for a VNF deployment. An orchestration service operating under our reference scenario could provide the necessary intelligence to consider these factors and improve energy efficiency of SUAVs. For instance, an assigning critical VNFs, which should have extended operation times, to SUAVs with longer battery lifetime. This should be done by the VIM according to the instructions provided by the NFV orchestrator. Estimating the residual battery of SUAV units is also fundamental to trigger the migration of VNFs and support effective re-allocation polices for them.

Another aspect that deserves careful consideration is the placement of SUAV units, for instance using GPS coordinates. In our reference scenario, NFVI nodes are mobile units that can be positioned at specific locations. This creates new challenges that are not present in traditional virtualization platforms, where NFVI nodes are interconnected through a high-speed fixed network (e.g. Gigabit Ethernet). On the contrary, in our reference scenario, the target position or the flight trajectories (e.g., in the form of waypoints) of each SUAV should be provided to the flight control engine running at the SUAV. Authors in work [6] proposed a solution where the flight control engine of each SUAV is implemented as a VNF, and the flight trajectories of the SUAV are provided to this VNF by a VNFM entity as configuration parameters. Following an alternative approach, the position or flight trajectories of the SUAVs could be specified as deployment options to the NFVO, or even be included in the NFV descriptor of the NS to be deployed. These parameters could then be provided by the NFVO to a specialized VIM, capable of configuring this information in the involved SUAV units during the deployment of the NS.

IV. PRACTICAL EVALUATION

After describing our theoretical analysis of the challenges and potential issues of NFV service orchestration in scenarios with intermittently available SUAVs, in this section we carry out a first evaluation of these challenges and issues from a practical perspective, using a well-known open-source software implementation of a VIM.

Our experimental setup is shown in Fig. 2. It has three aerial vehicles DJI Phantom 3 [21], each carrying a single board computer Raspberry Pi model 3 running a Linux distribution. In our experiments, the single board computers act as NFVI nodes, each of them being configured as a compute node that supports container virtualization under the control of an OpenStack Ocata VIM. This VIM runs on a mini-ITX computer (2.3 GHz processor, 16 GB memory, 128 GB hard drive, 4 GbE ports, and WiFi network card). The Raspberry Pis and the computer build a Wi-Fi Ad-hoc network (the network topology is shown in Fig. 2).

In principle, OpenStack does not provide the functionality to monitor the lifetime battery-powered compute nodes. It uses HTTP over TCP to support control operations over the NFV infrastructure resources. It does not support either the types of enhancement policies for the placement of VNFs that the described in the previous section. We divided our tests into two sets: the first set of experiments aimed at gaining an understanding of the effectiveness of OpenStack to deploy virtualized functions in situations of intermittent availability of control communications; a second set of experiments was designed to estimate the energy consumption produced by the orchestration and monitoring activities triggered by OpenStack at the NFVI nodes.

In order to completely understand the OpenStack behavior, we decided to cover the relevant cases that might be problematic during the orchestration process by capturing all traffic between OpenStack and the wireless nodes. In a first experiment, we captured all the traffic exchanged when an NFVI node is switched on and the OpenStack services are activated. There were two relevant traffic types exchanged during the experiment: HTTP requests related with status updates; and traffic corresponding to Advanced Message Queueing Protocol (AMQP) [19] Remote procedure Calls (RPC) triggered by the VIM. The latter required the establishment of 14 TCP connections, all associated to the same source port from the VIM equipment but different for the SUAVs.

The motivation for these TCP connections is related with the way OpenStack VIM handles the communication with the

OpenStack services running at the NFVI nodes: it uses the AMQP protocol to handle this communication through RabbitMQ, an open-source message broker [20]. Basically, instead of issuing HTTP requests, OpenStack relies on RPC to send and/or get data from the nodes using either an AMQP pull method, where OpenStack expects an answer from the node, or a push method, where OpenStack does not expect an answer. To manage effectively these methods, AMQP uses queues for every service on OpenStack, which in turn use dedicated TCP connections. Once the OpenStack services are activated, and the system enters into a stationary state, the exchange of HTTP and AMQP traffic occurs periodically, in background, at a lower rate. The number of TCP connections is always the same, even though the services might not be using them during a relatively long period of time. Therefore, some connections are kept alive when background traffic is exchanged.

To test behavior of OpenStack in situations of intermittent availability of control communications, we disabled the SUAV that acts as the point of contact with the VIM (SUAV 1) for 8 seconds, emulating a short interruption of the communications. In this case, as the failure situation is short, TCP's retransmission mechanism effectively handles the traffic lost, and the background traffic is re-established normally. To verify the VIM behavior in a long-term failure, we increased the interruption time to 10 min, to force all the possible retransmission timeouts to expire and close the TCP connections. After enabling back the SUAV, OpenStack is capable to re-establish the necessary TCP connections with the SUAV and resume the background traffic (it keeps in memory the status of the AMQP queues). Finally, we emulated a fairly small interruption of one minute and 10 seconds, to check what happens when not all TCP connections are closed during short failures. We observed that sender uses heartbeat messages to check if connections are still alive in AMQP, which basically behaves like TCP Keepalives. If receiver either answered the heartbeat or sent any kind of traffic through this connection, OpenStack would leave the connection alive. Otherwise, it would be closed. In this case, only some connections were still alive, but others had to be brought back online as if they were new connections.

Our next experiment aimed at investigating the VIM behavior during the deployment of a VNF in a SUAV. OpenStack requests the deployment of the VNFs using HTTP, and not AMQP. To evaluate the operation of OpenStack within situations of intermittent availability of network communications, we carried out the deployment of the VNF on SUAV1, disabling the SUAV for 8 s, 1 min 10 s, and 10 min. In all cases the deployment was eventually successful. A longer interruption delay of 15 min makes the deployment fail. This could be related with the HTTP session timer expiring at a time interval between the 10th and 15th minute.

To further test OpenStack ability to recognize failures and react accordingly, we tried to deploy a VNF while SUAV1 was offline. In long-term disconnections, after 10 minutes, OpenStack produces a deployment error, i.e., OpenStack does not allow deployments while network connectivity with the SUAV is considered unavailable. However, if the deployment is done shortly after the failure is emulated (1 min in our experiment), then OpenStack includes the deployment request in a queue delaying its execution.

Therefore, we can conclude that OpenStack seems to react fairly similar to the expected behavior we described in our analysis of the previous section, with respect to intermittently available control communications

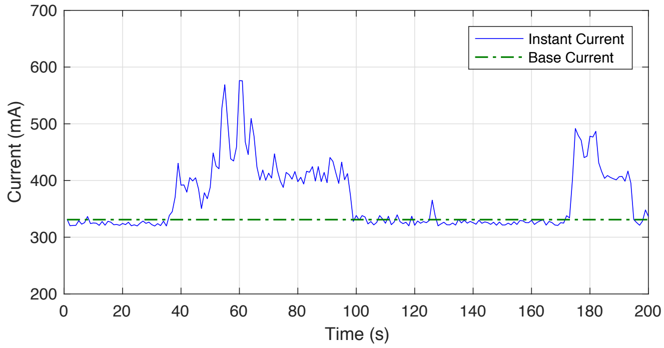

Fig. 3. SUAV network use cases.

Finally, we carried out a number of experiments to estimate the energy consumption at the NFVI nodes, resulting from the execution of the OpenStack services. With this purpose, we measured energy consumption in both SUAV2 and SUAV3 in the following cases: 1) before activating the OpenStack services at each SUAV (to obtain SUAV's base consumption); 2) after activating the OpenStack services at each SUAV and converging to a stable state, 3) after the failure of a neighboring SUAV; 4) during the failure of a neighboring SUAV, while a VNF deployment is carried out and reactivating the neighboring SUAV after a short period of time (2 minutes); and 5) in the same situation as in the previous step but considering a longer period of time for the reactivation of the neighboring SUAV (15 minutes).

In the first scenario, both Raspberry Pis showed a mostly stable power consumption, with an average value of 1389 mW. Once the OpenStack services were activated (second scenario), the average power consumption showed a very small increase of 9 mW (lower that 0.7%), reaching an average value of 1398 mW. Hence, it can be seen that background control traffic using AMQP does not heavily influence the battery consumption.

Further scenarios were compared using power consumption variations while disconnecting and reconnecting NFVI nodes, in order to check possible consumption peaks or the presence of any anomalous behavior. The third scenario had a total duration of 150 seconds, disabling SUAV1 at the 30th second. Disabling the SUAV did not have a noticeable impact on the energy consumption of SUAV2, and variations were almost neglectable (as seen in the previous two tests). The results corresponding to the fourth scenario are depicted in Fig. 3. In this case, while VNF is deployed at SUAV2, SUAV1 is disabled for approximately 1 min. As it can be observed in the picture, the energy consumption at SUAV2 increases during the deployment of the VNF, as expected. Right after SUAV1 is disabled, SUAV2 enters in a stable state and the energy consumption falls to the same value as scenario 2. When SUAV1 is re-enabled, the deployment of the VNF is resumed, and the energy consumption increases again. Finally, we repeated this scenario, but this time re-enabling SUAV1 after 15 minutes. In this case, the VNF cannot be deployed and the energy consumption remains at the same level of scenario 2.

V. CONCLUSION

Management and orchestration of infrastructure resources and virtual functions are fundamental to coordinate the operation of NFV environments. The introduction of new technologies in the telecommunications market, such as SUAVs, creates new opportunities for the fast and cost-effective deployment of network services following the novel

NFV paradigm. However, this opportunity opens new challenges and hurdles to NFV orchestration: 1) the limited lifetime of NVFI nodes, which has to be taken into account for migrating nodes and functions in replacement cases; 2) intermittent availability of control communications, which can make nodes unavailable for communications in short periods of time; 3) limited-capacity of NFVI nodes, which affects the load an SUAV is able to carry and the protocol used for exchanging information; 4) transport-layer protocols for control communications, whose performance may be decreases in mobile Ad-hoc networks, as in the case of TCP; and 5) supporting enhanced policies for VNF placement, which current VIMs do not provide as they are not aware of battery constraints and SUAVs location.

As a first practical application of our analysis, we deployed a simple scenario with a set of SUAVs, conforming a multi-hop wireless Ad-hoc network, and a well-known and widely-adopted VIM solution, i.e., OpenStack. OpenStack does not monitor the lifetime battery of compute nodes. It does not support energy and location-aware placements policies of VNFs, and it encapsulates control communications using HTTP over TCP, which may be problematic in a wireless setup. Nevertheless, our findings suggest that OpenStack handles the intermittent availability of NFVI nodes relatively well, as it is able to deliver control traffic even under transient failures of the network paths. Regarding battery consumption, the monitoring of the infrastructure resources provided by the SUAVs does not impose a significant increase, as compared to the case where OpenStack services are disabled.

Our future work includes an in-depth analysis of the identified challenges, and the development of specific solutions to appropriately address them and realize the view where SUAVs support the automated deployment and the operation of NFV services.

VI. Acknoledgement

This article has been partially supported by the European H2020 5GRANGE project (grant agreement 777137), and by the 5GCity project (TEC2016-76795- C6-3-R) funded by the Spanish Ministry of Economy and Competitiveness.

References

[1] J.G. Andrews, S. Buzzi, W. Choi et al., "What will 5G be?", *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065-1082, 2014.

[2] B. Nogales, V. S. Aguero, I. Vidal and F. Valera. (2018). “Adaptable and Automated Small UAV Deployments via Virtualization”. Sensors, vol. 18, pp. 4116-4132, Nov. 23 2018

[3] European Telecommunications Standards Institute. “Network Functions Virtualization (NFV); Architectural Framework; Research Report ETSI GS NFV 002 V1.2.1;” European Telecommunications Standards Institute (ETSI): Sophia Antipolis, France, 2014.

[4] Stallings, W. (2015) *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. (1st edition), [On-line]. Available: https://proquest.safaribooksonline.com/9780134175478 [December 22, 2018].

[5] J. Suarez, I. Vidal, J. Garcia-Reinoso, F. Valera and A. Azcorra, "Exploring the use of RPAs as 5G points of presence," 2016 European Conference on Networks and Communications (EuCNC), Athens, 2016, pp. 27-31.

[6] B. Nogales, V. Sanchez-Aguero, I. Vidal, F. Valera, and J. Garcia-Reinoso. 2018. “A NFV system to support configurable and automated multi-UAV service deployments”. *In Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet'18)*. ACM, New York, NY, USA, 39-44.

[7] Rametta, C.; Schembra, G. “Designing a Softwarized Network Deployed on a Fleet of Drones for Rural Zone Monitoring”. *Future Internet*, 9, 8, March 2017.

[8] K. J. S. White, E. Denney, M. D. Knudson, A. K. Mamerides and D. P. Pezaros, "A programmable SDN+NFV-based architecture for UAV telemetry monitoring," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, 2017, pp. 522-527.

[9] A. Bujary, C.E. Palazzi. “FANET Application Scenarios and Mobility Models”, in *Proceedings. On the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Application*, 2017, pp. 43-46.

[10] Z. Shelby, K. Hartke, C. Bormann. “The Constrained Application Protocol (CoAP)”. RFC 7252, June 2014.

[11] OASIS. “Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1”. ISO/IEC 20922:2016, June 2016.

[12] J. Liu and S. Singh, “ATCP: TCP for mobile ad hoc networks,” IEEE Journal On Selected Areas In Communications, vol. 19, no. 7, pp. 1300-1315, 2001.

[13] E. Larsen, “TCP in MANETs - Challenges and Solutions,” Norwegian Defence Research Establishment (FFI), 2012.

[14] G. Holland, N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks", Wirel. Networks, vol. 8, pp. 275-288, 2002.

[15] S. Allwin Devaraj, R. Helen Vedanayagi Anita, J. Jennifer Christa, "Comparative analysis of random based mobility models using TCP variant in MANETs", 2014 International Conference on Communication and Network Technologies, pp. 324-329, 2014

[16] Z. Fu, X. Meng, S. Lu, "How bad TCP can perform in mobile ad hoc networks", *Proc. IEEE International Symposium on Computers and Communications (ISCC'02)*, July 2002.

[17] D. Johnson, Y. Hu and D. Maltz. “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4”. RFC 4728, February 2007.

[18] C. Perkins, E. Belding-Royer and S.Das. “Ad hoc On-Demand Distance Vector (AODV) Routing”. RFC 3561, July 203.

[19] OASIS. “Advanced Message Queuing Protocol (AMQP) v1.0 specification”.

[20] Openstack. “AMQP and NOVA”. Internet: https://docs.openstack.org/nova/latest/reference/rpc.html [Dec. 27, 2018].

[21] DJI. Phantom 3 Professional. Available online: https://www.dji.com/phantom-3-pro (accessed on January 2019).

[22] OpenStack. Open Source Software for Creating Private and Public Clouds. Available online: https://docs.openstack.org/ocata (accessed on January 2019).