Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 4734821, 20 pages
https://doi.org/10.1155/2019/4734821

Research Article

An NFV-Based Energy Scheduling Algorithm for a 5G Enabled Fleet of Programmable Unmanned Aerial Vehicles

Christian Tipantuña,[1,2] Xavier Hesselbach,[1] Victor Sánchez-Aguero,[3] Francisco Valera,[4] Ivan Vidal,[4] and Borja Nogales[4]

[1] *Department of Network Engineering, Universitat Politècnica de Catalunya, Spain*
[2] *Department of Electronics, Telecommunications and Information Networks, Escuela Politécnica Nacional, Ecuador*
[3] *IMDEA Networks Institute, Spain*
[4] *University Carlos III of Madrid, Spain*

Correspondence should be addressed to Christian Tipantuña; christian.jose.tipantuna@upc.edu

Received 16 November 2018; Revised 29 January 2019; Accepted 7 February 2019; Published 20 February 2019

Academic Editor: Giovanni Stea

The fifth generation of mobile networks (5G) is expected to provide diverse and stringent improvements such as greater connectivity, bandwidth, throughput, availability, improved coverage, and lower latency. Considering this, drones or Unmanned Aerial Vehicles (UAVs) and Internet of Things (IoT) devices are perfect examples of existing technology that can take advantage of the capabilities provided by 5G technology. In particular, UAVs are expected to be an important component of 5G networks implementations and support different communication requirements and applications. UAVs working together with 5G can potentially facilitate the deployment of standalone or complementary communications infrastructures, and, due to its rapid deployment, these solutions are suitable candidates to provide network services in emergency scenarios, natural disasters, and search and rescue missions. An important consideration in the deployment of a programmable drone fleet is to guarantee the reliability and performance of the services through consistent monitoring, control, and management scheme. In this regard, the Network Functions Virtualization (NFV) paradigm, a key technology within the 5G ecosystem, can be used to perform automation, management, and orchestration tasks. In addition, to ensure the coordination and reliability in the communications systems, considering that the UAVs have a finite lifetime and that eventually they must be replaced, a scheduling scheme is needed to guarantee the availability of services and efficient resource utilization. To this end, in this paper is presented an UAV scheduling scheme which leverages the potential offered by NFV. The proposed strategy, based on a brute-force search combinatorial algorithm, allows obtaining the optimal scheduling of UAVs in time, in order to efficiently deploy network services. Simulation results validate the performance of the proposed strategy, by providing the number of drones needed to meet certain levels of service availability. Furthermore, the strategy allows knowing the sequence of replacement of UAVs to ensure the optimal resource utilization.

1. Introduction

Recent evolution in Unmanned Aerial Vehicles (UAV) boosted by the miniaturization of electronic and sensors have allowed the use of UAVs in different civilian applications. Their shrinking size in combination with price reductions has increased the popularity of these devices both in the amateur community as well as in professional applications. Accordingly, we are now witnessing the fast deployment of a new categorization in the UAV area: Small Unmanned Aerial Vehicles (SUAV), commonly known as drones (that will be the preferred name in this article), which are low-cost devices with reduced payload capacities, restricted communication range, and limited battery time, but still powerful enough so as to carry small computers on board.

Drone applications are spreading throughout a plethora of different fields covering from smart agriculture scenarios to road traffic monitoring, public safety, sensor information retrieving, or even unmanned cargo. In general, these use cases are normally scheduled as relatively fixed missions of standalone drones [1]. This paper, in particular, is focused on the energy management challenge such that, although is obviously present in standalone drone short missions, its complexity is exponentially exacerbated when dealing with multidrone long-term operations.

This research work has been done within the framework of the Spanish research project 5G-City (5GCity is a coordinated national project (2017-2019), funded by the Spanish Ministry of Economy and Competitiveness with the following partners: Universidad Carlos III de Madrid, Universidad de Granada, Fundacio i2Cat, Universitat Politecnica de Catalunya, Universidad de Vigo, and Universidad del Pais Vasco) that focuses on the provisioning of solutions for unpredictable critical events such as natural disasters or crowded events that produce damage and faults in the conventional network infrastructure (i.e., legacy infrastructure needs to be reinforced or is simply not available). The use of multidrone network is apparently a cost-effective solution which enables a fast and agile deployment in hard-to-reach locations and can straightforwardly be integrated into existing networks and adapt to unexpected changes. This flexibility can be certainly improved with the usage of Network Function Virtualization (NFV) 5G technology enabled drones as we have shown in [2]. However, drones have also several challenges that should be addressed. The weight a drone can carry determines the payload equipment and the size of the on-board battery. In consequence, we deal with low-resource payload (e.g., Single Board Computers) equipment and small batteries that provide limited service time. Because of these limitations, we need multidrone systems to cover areas of significant size and a fleet of reserve drones for replacements in order to provide long-term services. Apart from connectivity requirements, such as latency and bandwidth, a communication system provided by drones needs innovative management solutions (e.g., NFV) that enable the use of resources and energy efficiently.

For this reason, this article presents a strategy for the efficient management of resources in a communications system that provides services or network functions through the deployment of drones. The proposed solution leverages the potential offered by NFV and the 5G capabilities. In the context of the proposal, 5G technology is used to meet connectivity requirements, such as very low latency and high bandwidth in order to guarantee a correct migration procedure and also to provide communication between the different components in the system. Instead, NFV is in charge of the management tasks in the system. Specifically, in order to carry out the management tasks related to the replacement of drones and allocation of drones to services, an energy-aware scheduling algorithm has been developed, which is the main contribution of this paper.

The proposed algorithm, based on a brute-force search combinatorial method, explores all possible combinations of drones and service with the aim of providing the exact or optimal scheduling of drones to execute services. This exact allocation of drones over time ensures the continuity of services during a finite time interval, while leading to the optimal resource utilization. Apart from the replacement sequence, the algorithm can inform the total number of drones (or batteries) to use to reach a certain level of availability. In addition, within an NFV scope, the drone scheduling strategy can be considered as a network service.

To validate the performance of our solution, two small-scale scenarios have been analyzed, one Generic and one Realistic, whose results can be applied in the planning of design stages in a variety of real use cases, such as services in emergency or natural disasters and relief services in search and rescue missions. In addition, the information obtained with our implementation is also useful as a baseline to develop mathematical models and faster suboptimal or heuristics methods for real-time practical implementations.

The rest of the article is organized as follows: Section 2 provides a general overview of the related work. In Section 3 the main problem is formally presented. Section 4 presents the drone scheduling procedure and the complexity analysis. The performance evaluation is described in Section 5 and results are illustrated in Section 5.2. Finally, Section 6 concludes the article.

2. Related Work

In the last years, drone uses have evolved from the basic on-board video camera applications to a wide range of novelty functions such as drones acting as first responders in an accident or drone swarming intelligence to provide network services. To conduct these assignments efficiently, on account of drones' limitations, the use of 5G technologies such as NFV or SDN seems essential as they will enable an accurate operation. In particular, in this article, NFV is used to exemplify the execution of the proposed algorithm. There are several examples of the use of NFV in the UAV domain in the literature. In [3] an UAV platform provides to external controllers the opportunity to adapt the telemetry monitoring. In [4] is presented an NFV programmable infrastructure that enables the agile unification of services and functions, which may be determined by the operator of the UAVs at deployment time. NFV is used to decouple the drone hardware infrastructure from the control layer that virtualizes the infrastructure resources for the higher layers [5]. NFV is also used to enable multimission drones and supports a flexible deployment of network services [2]. Finally, NFV allows the migration of Virtual Network Functions (VNF) [3], which are the responsible units for providing the network functionality through the software implementation. The VNF migration enables an agile and flexible execution of the network services encompassing those VNFs that can be accommodated by the drones. Basically, the migration of VNFs consists of moving a virtual machine from one drone unit to another. There are different migration types: nonlive migration, where the VNF is down and it is moved to a different compute node, and live migration, where the VNF is running throughout the migration. Well-known tools that are key in the NFV framework development, such as OpenStack (OpenStack: https://docs.openstack.org/ocata) or VMware (VMware: https://www.vmware.com/es.html), support migration. The use of NFV is reinforced by the appearance of multidrone systems. Drones can run different

VNFs, endowing a huge versatility to the drone swarm. Nonetheless, virtualization is a resource intensive process, and because of the limited on-board equipment, it is necessary to use solutions such as LXC Linux Containers (Linux Containers: https://linuxcontainers.org/) to provide a similar environment as a Virtual Machine (VM) but reduce the overhead that comes with running a separate kernel and simulate all the hardware. It should be noted that the migration of container-based VNFs presents an additional challenge since this virtualization unit is stateless and in principle cannot be migrated. However, there are recent works in which they aim at addressing migration issues with containers like CRIU (a project to implement checkpoint/restore functionality for Linux: https://github.com/checkpoint-restore/criu). For this reason, in this article, because of the selected VNFs, the migration process is not mandatory. Routing VNFs recover their status proactively, collecting all the necessary routes in a few seconds but, in complex network scenarios, the use of VNF migration is crucial for correct operation.

Different alternatives for drone communication have been proposed in [6] being the WiFi in ad hoc mode one of the most popular solutions. Regarding routing strategies, there are also several options that extend from mobile ad hoc networks (MANETs) and vehicular ad hoc networks (VANETs) and also some innovative proposal like Software Defined Network (SDN).

The main focus of this paper is set on the battery power consumption. All the drones are normally equipped with a Single Board Computer as payload (Raspberry Pi 3B (Raspberry Pi 3B: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/) (RPi) in our case), with an autonomous battery giving rise to an independent service of the drone. In standard drone applications (drone flying as expected), flight-engines consume most of the energy [7] while the power consumption by network services is practically negligible, so that the service time is limited by the drone battery time (around 20 minutes following the technical specifications (DJI Phantom 3 Pro: https://www.dji.com/es/phantom-3-pro)). Even so, when a drone has a static position, it tends to land whenever possible to save energy (a drone that is providing a WiFi access point service does not necessarily have to be flying). In this case, service time will be limited by the SBC battery time and network services should be taken into account and will play an important role in modeling battery consumption.

Regarding power consumption in mobile and portable devices, there are different examples studying the impact of hardware components on the energy consumption [8, 9] and also the impact related to wireless communication [10]. In [11] is presented a method for wirelessly charging the drone battery when it lands, without the need to remove it and replace it. Ground task automation has come to the attention of researchers during the past few years [12, 13] reducing the human operators at the Ground Control Station (GCS).

In addition, in order to efficiently manage the available resources (e.g., energy), various techniques, mechanisms, and procedures have been developed. One of the most widely used is the combinatorial analysis, in which all possible combinations of resources to be used are analyzed. In this proposal, this mechanism is used to analyze all possible combinations of drones to run services. Considering a procedure similar to that described in [14], the proposed technique, by analyzing the whole set of possible cases, ensures the best (exact) result by providing the information of specific resources (drones and batteries) to be used. This optimal scheduling of drones guarantees an efficient use and management of available energy at every moment.

3. Problem Statement and System Model

In this section, first the statement of the problem is formally presented in Section 3.1. Then, the system model and notations are described in Section 3.2 followed by the definitions in Section 3.3 and the performance metrics in Section 3.4. Finally, the assumptions are presented in Section 3.5.

3.1. Problem Statement. Maintaining a certain degree or level of availability can become an important and even critical consideration in the deployment of network services. Especially in communication systems provided by drones, whose capacities in terms of processing and energy may have limitations, the efficient use and management of resources must be guaranteed in order to provide or maintain a desired level of availability. Therefore, this metric is an important factor in the design, planning, and deployment phases, considering that some applications may demand specific values for their operation.

In order to provide network services, by leveraging the connectivity capabilities offered by 5G networks and within an NFV context, a set of programmable drones can run VNFs, and, thus, provide the required services. In this sense, a fleet of programmable drones can offer different network services simultaneously, such as routing tasks, Internet connectivity, video surveillance services, telemetry, and multimedia services. To ensure proper coordination and management of the devices that implement the VNFs or services, it is necessary for an entity or component to perform the corresponding management tasks. In this way, and in an NFV environment, the core management entity, i.e., the orchestrator, can perform the orchestration and management of available resources [15].

Besides, because the provision of services provided by drones is constrained to their autonomy or battery duration, an efficient energy management scheme is of paramount importance in both short- and long-term applications. In this regard, a policy or scheme that allows the coordination and replacement of drones, to keep the service in an active state while ensuring a certain level of availability, is essential. As a result of all aforementioned, this work presents a scheme or management system for the deployment and replacement of drones, in which an optimal scheduling algorithm is implemented in order to guarantee the continuity of services, i.e., a level of availability, during a finite time interval.

The proposed scheme is shown in Figure 1 and is composed of two components: (i) a set of drones, which are in charge of executing the VNFs and that constitute the Network Function Virtualization Infrastructure (NFVI)

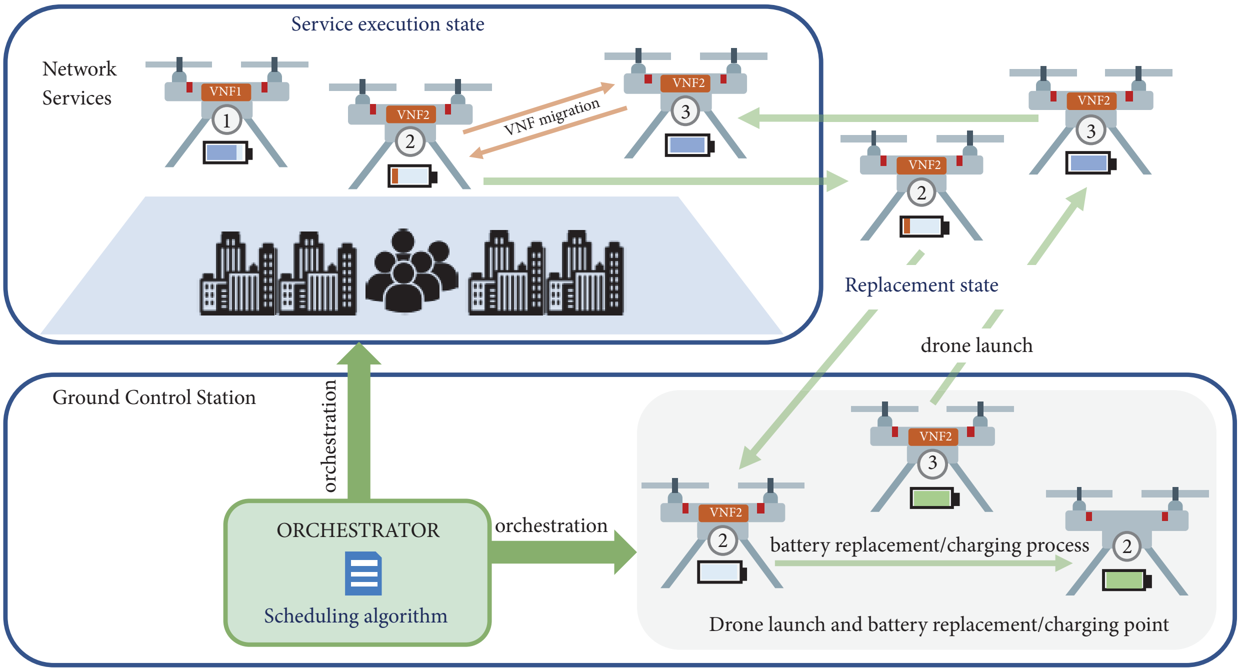


Figure 1: Overview of the proposed approach.

and (ii) a GCS, where the elements and entities responsible for monitoring, control, and management of resources and network services are located. The latter is precisely the component (NFV orchestrator) where the developed algorithm is intended to be executed. In this regard, the complete system can be considered as a network service, which in an NFV environment and using the connectivity benefits provided by 5G such as very low latency and high bandwidth can offer the optimal or exact drone scheduling for services execution. In addition, because 5G is considered in the design of the system, the proposed solution can also be categorized as a novel 5G use case.

The goal of the proposed algorithm is to carry out an optimal drone scheduling over time, in order to maximize the use of available resources, drones, while providing a reliable communications system guaranteeing continuity in the execution of services. In addition, the information provided by the algorithm can be used as a toolkit in mission planning. Apart from the replacement sequence, the algorithm can inform the services availability level obtained with the deployment of a given number of drones, or in turn the results can be used to know the number of drones that must be deployed to obtain a given service level.

The proposed scheme is characterized based on two different states, which are described as follows.

(1) Service Execution State. In this state the drones, which are equipped with processing and communication devices, execute the VNFs to provide the demanded services. For its operation, the drones are battery powered. Therefore, the autonomy time or drone lifetime is constrained to the capacity of the power supply, the energy consumption of the services to be executed, and the consumption of all the elements that allow the operation of the device.

In the proposed approach, the drones can execute the VNFs or services while they are in flight, as shown in the example of Figure 1. Also, for strategic reasons and with the aim of extending the service lifetime, it is possible to consider scenarios in which not all drones remain in flight. For example, for certain applications, such as the provision of connectivity services, some drones after their launch may land on specific locations. In the latter scenario, the service provided by the drones on the ground is not limited to the time that the drone can remain in the air. Even in this case, it is possible to consider the use of a secondary energy source to further lengthen the time of service provision. In any of the proposed scenarios, flying drones or drones on land, the algorithm guarantees the optimal drone scheduling overtime over time. Of course, depending on the scenario, for example, if all the drones are flying, the drone replacement procedure should be performed more frequently.

(2) Replacement State. In this state the drones do not provide the services. However, this phase is necessary to guarantee both the migration (transition) of VNFs and the replacement or recharging of drone batteries. Since the battery duration has a finite lifetime, it must be recharged or replaced, the replacement being a more useful and practical option in most cases, due to the agility involved in the process.

In the proposal, the management system located at GCS has all information about available resources (drones and batteries) and service requirements (power demanded by each service and the total required availability time) because all is provided by the users of the system. Through the execution of a scheduling algorithm, the system is able to provide the optimal allocation of drones to cover the demanded services. The scheduling algorithm is executed in the NFV domain, specifically in the NFV orchestrator as depicted in Figure 1.

For the computation of the optimal allocation of drones, the algorithm considers both the consumption related to the service (*service execution state*) and the necessary power to perform the replacement process (*replacement state*). In the proposal, these power consumptions are represented by time variables, a *service time* related to the *service state* and a *replacement time* associated with the *replacement state* (see Figure 3). Thus, a percentage of the total battery capacity (majority) is assigned to service execution and the rest is dedicated to the execution of the replacement tasks (service migration). The time variables associated with the operating states of the system are described in detail in Section 3.3. Regarding the *service time*, this value depends on the power demanded by the service and can vary greatly from one service to another; for example, considering the same battery capacity, a drone running a service that demands high power consumption will have a shorter *service time* compared to another running a service whose power consumption is lower. On the other hand, the *replacement time* is composed of the time needed to perform the migration of the VNFs, from old drone (low battery level) to new drone (high battery level), and the time associated with the round trip flight to (old drone) and from (new drone) the GCS. Since the replacement time is not directly linked to the execution of the service, the value of this parameter could be similar or the same for the different services.

In order to guarantee the continuity of the services in the proposed system, the service migration process starts when the drone that is going to be replaced is active; i.e., when it is running the service, specifically, the migration process begins at the end of the *service time* or at the beginning of the *replacement time* (for a better understanding see Figures 3 and 5(e)). After the migration process has been carried out, i.e., when the *replacement time* is over, the replaced drone returns to the GCS; at this time this drone is no longer active but is still part of the system. Once the drone reaches the ground, its battery is replaced or recharged so that, according to the indications received by the GCS, it can be assigned for the execution of another service.

According to the aforementioned, in the system, the replacement time is sufficient to guarantee the transition of the services as well as the launching and landing of the drones. In addition, regarding the migration process, among the important aspects to consider are the service hand-off processes, from one drone to another, and the exchange of information associated with this procedure. Regarding the latter, in the proposal the exchange of information is accomplished thanks to features such as high connectivity and low latency time provided by 5G technology. Instead, the procedure related to the service transition is a process linked to the type and features of each service; therefore, although this is an interesting topic, it is not addressed in the article since it is out of scope of the proposal. However, it is worth mentioning that Section 5 presents the results of the application of the proposal in a real case whose values of both the *service state* and the *replacement state* (including the migration process) have been obtained through measurements.

At all times, the management system coordinates the resources that must be allocated (drones to be launched from the ground), because based on the initial information of services and drones, as well as the computations performed by the algorithm, the system is able to estimate the number of available drones, the status of the services, the sequence of replacement to be performed, and the availability level reached. Hence, the characterization of the system through the *service state*, the *replacement state,* and their corresponding time variables enables the system to operate with the appropriate margins so that the services can be executed continuously during a required time interval while the resource utilization is optimized.

For a better description and understanding of the different states of the proposed energy management scheme, an example is presented below. In Figure 1 is considered an application environment composed by two VNFs, which are expected to be active during a finite time interval. To this end, the system initially uses two drones, drone 1 and drone 2, which execute VNF1 and VNF2, respectively. As time goes by, the management system evidences that drone 2 is draining its battery due to the consumption of the service and the consumption related to its flight. In response to this, and before the drone stops providing the service or in the worst case it stops working and collapses to ground, the system coordinates the sending of another drone. In this case drone 3 is selected, whose energy level is adequate to guarantee the execution of the VNF 2 for a subsequent time interval. At the moment that drone 3 is located at a suitable distance for the establishment of communication with drone 2, the migration of VNF2 from drone 2 to drone 3 is performed, so that the service is not interrupted and remains available. Subsequently, drone 2 returns to the ground station to recharge or replace its battery, so that it can be ready for a new allocation. Thus, drone 2 is available to run the VNF2 or a different VNF, and it depends on the decision that is made by the scheduling algorithm and the corresponding management system.

During all the time of operation of the service, all the actions both on land and in the air are coordinated by the management and orchestration systems. In summary, the replacement state includes the launching of the new drone (with high battery level) from the ground station, the return of the old drone (with low battery level) to the ground station, and the service migration process (VNF migration).

In addition, from the example described above, it can be observed that to guarantee a continuous execution of the service and a total availability level (100%), the number of available drones must be at least one unit greater than the number of services. In the example it is verified that, to guarantee the continuous operation of VNF1 and VNF2, it is necessary to use 3 drones, drone 1 (VNF1), drone 2 (VNF2), and drone 3 (VNF2).

Also, as aforementioned, the replacement state may include the battery replacement or recharge of it. In the first

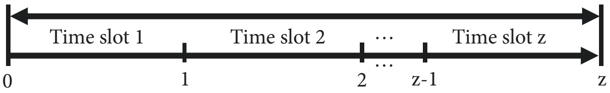


Figure 2: Time representation.

case, the battery replacement is a process that can generally take less time; for example, in a search or rescue mission, it is possible to use a limited number of drones and a large number of batteries. Meanwhile, in the second case, the battery charging commonly is a slower process, but necessary if the drone is tampering resistant, or if the number of available batteries is limited.

In the proposal, regarding the replacement state, for practical reasons, has been considered the battery replacement procedure. Nonetheless, the algorithm developed has the flexibility to consider a battery charging procedure. In fact, within the characterization of the system, the battery charging phase could be considered as an additional state, the *battery charging state*.

In summary, the proposed strategy bases its operation on a drone scheduling algorithm, which allows knowing how many drones are going to be used, how they should be replaced, and when the replacement should be made.

3.2. System Model. The drone scheduling algorithm is intended for providing the information of the optimal drone scheduling over time. In the proposal, the time variable has been divided into time slots, as shown in Figure 2. Thus, a drone is able to run a service over time (service execution state) during one or several slots according to its capabilities (available energy) and the features of the services that it can run. Similarly, the drone replacement state can last one or more time slots depending on the features of the drones (battery replacement procedure) and the scope of application of network services.

A summary of notations that describe the drone scheduling strategy is shown in Table 1. Then, these parameters are defined in Section 3.3.

3.3. Definitions

(1) *Expected availability time* (T_A^E): also defined as service availability time, it represents the time interval where the services are expected to be active/available.

(2) *Reached availability time* (T_A^R): time interval during which the services are active/available.

(3) *Number of services* (NS): the set of services or VNFs that are executed by the drones during a certain time period.

(4) *Initial time of service* j (T_{init}^j): time instant from which the service j is available/active, initial time in which the service availability is analyzed.

(5) *Power demanded by the service* j (P_d^j): power demanded by each service j to be executed. Each service may demand a different amount of power. In general, this parameter represents the consumption demanded by the services, and in practical implementations its units can also be given in terms of electric current (e.g., [mA]).

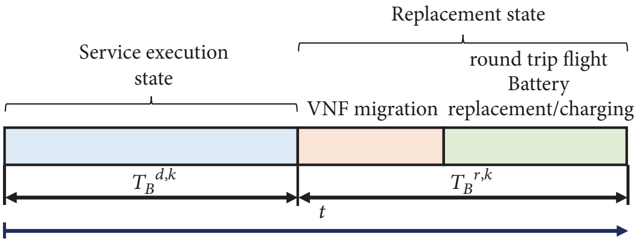


Figure 3: Time variables of the drone scheduling strategy.

(6) *Number of drones available* (ND): set of drones that are part of the system.

(7) *Battery capacity of drone* k (C_B^k): it represents the amount of energy that can be stored in a battery of each drone k. Moreover, in practical implementations this capacity can be expressed in terms of electric charge, i.e., electric current per time units (e.g., [mAh]).

(8) *Drone battery lifetime* ($T_B^{d,k}(P_d^j)$): each drone k with a battery capacity (C_B^k) can execute a service that demands a power level (P_d^j) during a time period $T_B^{d,k}$. This relationship can be expressed as follows:

$$T_B^{d,k}\left(P_d^j\right) = \frac{C_B^k}{P_d^j} \tag{1}$$

This time variable represents the time interval linked to the *service execution state*.

(9) *Battery replacement time* ($T_B^{r,k}$): this time variable is linked to the *replacement state* of a drone k. The ($T_B^{r,k}$) includes the time associated with the sending of the new drone (drone with high level of energy supply), the time demanded to perform the migration process of the services, and the time needed for the old drone (drone with low level of energy supply) to reach the ground station (charging point).

A pictorial representation of the time variables related to the two states that characterize the system is shown in Figure 3.

3.4. Metrics. To assess the performance of the drone scheduling algorithm, two metrics have been defined.

(1) *Services availability* (A_v): also defined as the total availability of services and expressed as a percentage, this metric shows the ratio between the time that all the services are available and the expected availability time. If the (T_A^E) = (T_A^R), i.e., all the services are available during all the time required, the (A_v) = 100

Table 1: System parameters.

Parameter	Description	Comments/Units
T_A^E	Expected availability time	Time units
T_A^R	Reached availability time	Time units
A_v	Service availability	Percentage, $A_v \in \{0, \ldots, 100\}$
A_vS	Service availability per services	Percentage, $A_vS \in \{0, \ldots, 100\}$
NS	Number of services (VNFs)	Integer number
S_j	Service identifier	$j \in \{1, \ldots, NS\}$
T_{init}^j	Initial time of service j	Time units
P_d^j	Power demanded by the service j	Power units
ND	Number of available drones	Integer number
D_k	Drone identifier	$k \in \{1, \ldots, ND\}$
C_B^k	Battery capacity of drone k	Power x Time units
$T_B^{d,k}(P_d^j)$	Battery lifetime of drone k for service j	Time units
$T_B^{r,k}$	Battery replacement time of drone k	Time units

%; otherwise, this value will be lower. The service availability can be expressed as

$$A_v = \frac{T_A^R}{T_A^E} \cdot 100\% \tag{2}$$

(2) *Services availability per services* (A_vS): this metric provides the information of the mean availability value of all services. A service j can reach an availability level equal to $A_{v,j}$, if this value is small compared to the availability of the other services, the A_v value will also be small and equal to $A_{v,j}$. For this reason, the (A_vS) metric is defined, because it is less restrictive and weights all availability values, in order to provide information on the behavior of all the services that are part of the system. As a consequence, the A_vS value will always be greater or at most equal to the A_v value.

The service availability per services is defined by

$$A_vS = \frac{\sum_{j=1}^{NS} A_{v,j}}{NS} \cdot 100\% \tag{3}$$

3.5. Assumptions. The following assumptions are made for the practical implementation of the algorithm:

(1) In practical implementations each programmable drone can execute more than one VNF concurrently. However, to simplify the analysis, in the proposed scheduling scheme, each drone k can run only one VNF or service j. This consideration is valid, since the execution of several services in the same drone would correspond to the consumption of different power levels. Thus, the processing of only one VNF and the analysis of its consumption could represent a summarized value of all the services that are executed in the drone.

(2) Similar to the previous consideration, the strategy considers that a service j can only be executed by one drone k at the same time. This is with the aim of simplifying the analysis in the distribution of drones and services.

(3) In the proposed system, any drone has the ability to execute any VNF. Likewise, any battery can power any available drone. In this sense, all available resources, drones and batteries, can be reused when demanded. It is clear that the services execution is limited to the capabilities of drones and the features of services, as previously discussed in Section 3.1.

(4) In the proposal it is considered that all services work simultaneously, i.e., all services are available as long as the system has the resources for their execution.

(5) The ND must be at least equal to NS. However, as discussed in Section 3.1, to ensure the execution of services without interruption, ND should be at least greater than or equal to $NS + 1$ ($NS \geq ND + 1$). If $ND < NS$, then $A_v = 0\%$ and $A_vS = 0\%$. The aforementioned consideration is mandatory in the initial execution or first drone allocation process, after this stage the algorithm is continuously evaluating the amount of available resources. Therefore, in the following allocation processes ND could be smaller than NS, in which case the algorithm analyzes the requirements of services to perform the corresponding allocations.

(6) In the *replacement state*, the drone that is replaced arrives at the ground station, with a very low battery level (fully discharged battery). For the replacement process, a battery that has previously been charged up to 100% of its capacity is used (fully charged battery). Similarly, if the complete drone must be replaced and not just its battery, the device that replaces it will be equipped with a battery charged to the maximum level. This consideration is also valid for the drones that are assigned for the first time; i.e., the drones used in the first allocation process have their batteries fully charged. In addition, the system has enough batteries to guarantee the replacement process of all the drones that demand them.

(7) In the proposal it is assumed that communication requirements such as very low latency and high bandwidth capabilities are provided by 5G technology. Moreover, the level of connectivity provided by 5G allows for proper communication and coordination between the different components within the system.

4. Drone Scheduling Strategy

In this section, first the drone scheduling procedure is presented in Section 4.1; then, the complexity of the problem is discussed in Section 4.2.

4.1. Drone Scheduling Algorithm Procedure. The drone scheduling strategy consists of systematically computing the optimal set of available of drones to execute the services. To this end, the strategy follows the guidelines described in the *execution* and *replacement* states.

The scheduling process starts with the individual analysis of the execution of each service for each available drone ($T_B^{d,k}(P_d^j)$) then goes through the following three phases to obtain the optimal allocation of drones to run services.

(a) *Computation of Combinations.* In order to find the exact or optimal allocation of drones to run services, the algorithm, based on a brute-force search combinatorial method, explores all possible combinations of drones and services. Once all the possible combinations are obtained, the algorithm determines those that meet the system requirements (valid combinations). Subsequently, this set of combinations is sorted in descending order according to the A_v metric. At the end of this phase, the best combination of drones (the first combination in the list from the top) is selected. In this context, this best combination represents the optimal set of drones whose services execution produces the highest A_v value in the system.

(b) *Resource Allocation and Services Evaluation.* Once the best combination of drones and services is obtained, the drones are allocated to their corresponding services. Afterwards, the A_v and A_vS metrics are computed; specifically, the A_vS metric is computed because the A_v metric was already obtained in the previous phase. If the A_v reached is equal or greater than the desired value, i.e., $T_A^R \geq T_A^E$, the algorithm stops its execution; otherwise, it analyzes the current availability level of all services ($A_{v,j}$) and the available resources (drones that have not been used) to proceed with the next allocations.

The analysis of the available resources is carried out in the following phase, while the analysis of availability per services is part of this phase and corresponds to the services evaluation, which is a procedure performed in order to reach the highest possible A_v or T_A^R value (both parameters completely equivalent), from the second allocation process. In this regard, based on the information of the last allocation made, the algorithm lists the services in descending order according to the $A_{v,j}$ reached, so that this information can be used in the computation and subsequent allocation of the best combination of drones. In specific, the objective of this process is to provide additional information to the algorithm in order to allocate the drones with the highest battery capacity to the services with the lowest current $A_{v,j}$ values. In summary, the services evaluation contributes that the drones are allocated starting with the service with the lowest $A_{v,j}$ value. The mechanism described in this phase ensures an increasing A_v and efficient resource utilization.

(c) *Verification of Available Resources.* Throughout the scheduling process, the algorithm must know the status of the executed services (T_A^R) and the information of the resources in the system. Especially from the second allocation process, the algorithm has to identify the resources used and available in order to perform the computation of combinations and the subsequent allocation of resources. In this regard, the algorithm has to evaluate at each time the information of the drones in the system, considering that this information consists of drones used, drones that have not been used, drones that must replace their battery, and drones whose battery has been replaced and are ready for a new allocation.

In an iterative process, the algorithm follows the phases described above and continuously calculates the best scheduling of drones to execute services. This procedure is carried out constantly until any of the two stopping criteria is met. The first criterion is the A_v value reached; if after an allocation process $A_v = 100\%$, the algorithm stops its execution. The second stop criterion is related to the number of available drones in the system, considering that this number is made up of drones that have not been used (not allocated yet) and drones whose battery has been replaced (or charged). In the event that the system does not have the necessary resources (drones) to perform the corresponding allocations, the algorithm stops its execution, under this condition $A_v \neq 100\%$ will be achieved. Finally, the algorithm provides the information of the (T_A^R), A_v, and A_vS reached.

The developed algorithm guarantees the best drone scheduling for services execution over time, by analyzing all possible drones-services combinations. However, the problem tends to growth as the *NS* and *ND* increase, which can be a problem if the capacity or processing time are constrains within the system.

The phases discussed above are implemented in the algorithm through different steps. The drone scheduling algorithm is explained in Figure 4 and each step is described in detail based on the example depicted in Figure 5. In this example $T_A^E = 7$ [time slots], and, for simplicity, the *NS* and *ND* are limited to 2 and 4, respectively. A pictorial representation of the required services is shown in Figure 5(a). Moreover, the example considers a $T_B^{r,k} = 2$ [time slots]; the first time slot corresponded to the VNF

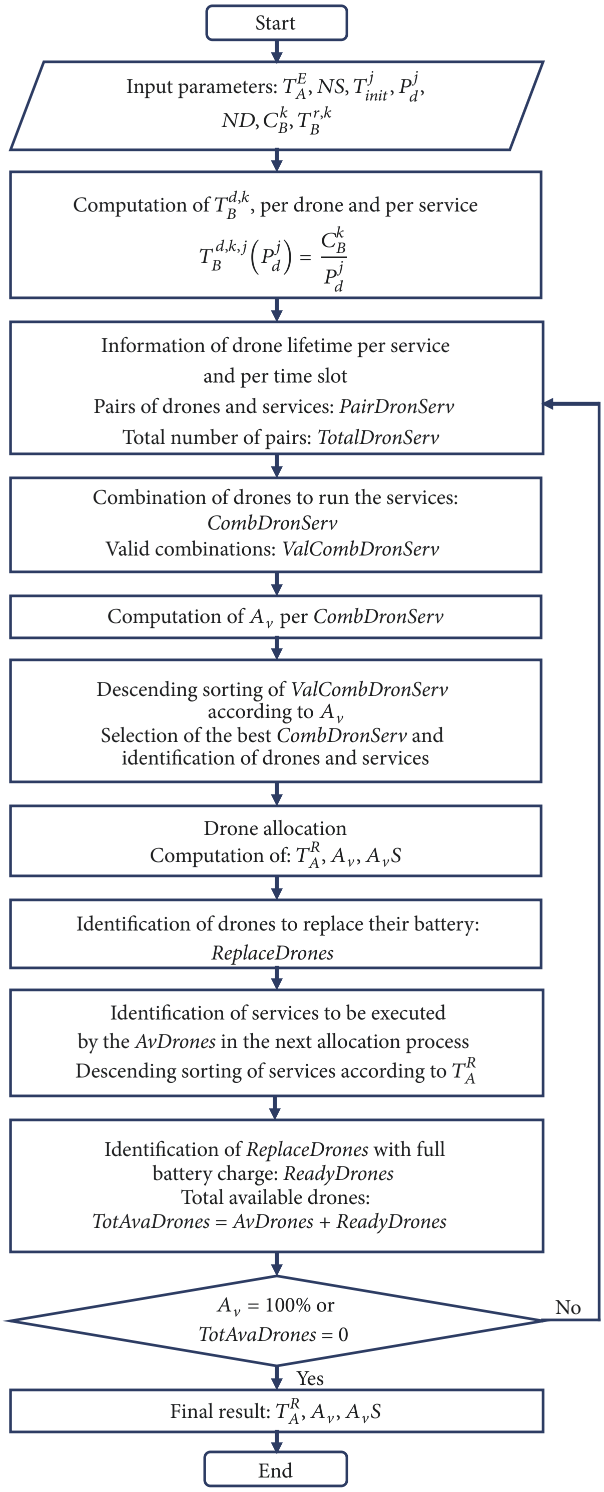

Start
Input parameters: T_A^E, NS, T_{init}^j, P_d^j, ND, C_B^k, $T_B^{r,k}$
Computation of $T_B^{d,k}$, per drone and per service
$T_B^{d,k,j}\left(P_d^j\right) = \frac{C_B^k}{P_d^j}$
Information of drone lifetime per service and per time slot
Pairs of drones and services: *PairDronServ*
Total number of pairs: *TotalDronServ*
Combination of drones to run the services: *CombDronServ*
Valid combinations: *ValCombDronServ*
Computation of A_v per *CombDronServ*
Descending sorting of *ValCombDronServ* according to A_v
Selection of the best *CombDronServ* and identification of drones and services
Drone allocation
Computation of: T_A^R, A_v, A_vS
Identification of drones to replace their battery: *ReplaceDrones*
Identification of services to be executed by the *AvDrones* in the next allocation process
Descending sorting of services according to T_A^R
Identification of *ReplaceDrones* with full battery charge: *ReadyDrones*
Total available drones:
TotAvaDrones = *AvDrones* + *ReadyDrones*
A_v = 100% or *TotAvaDrones* = 0
No
Yes
Final result: T_A^R, A_v, A_vS
End

Figure 4: Algorithm for drone scheduling.

migration process and the round trip time of the drone, and the second time slot referred to the time for battery replacement. In addition, to better understand the proposed strategy in Table 2 is provided a summary of parameters related to the processing of the combinations and the analysis of the drones in the system. The different steps that are part of the algorithm are explained as follows.

(1) Input parameters: the input parameters comprise the T_A^E value and the information of services and drones, as shown in Tables 3(a) and 3(b), respectively.

(2) Computation of $T_B^{d,k}(P_d^j)$, per drone and per service: for the computation of these values, (1) is used.Table 4(a) presents all possible values of battery lifetime per drones $(D_1, \ldots, D_4)$ and per services (S_1, S_2).

(3) Information of drone lifetime per service and per time slot: in this step is provided the same information displayed in Table 4(a) but disaggregated in pairs of drones and services ($PairDronServ$), as shown in Table 4(b). Further, in this step the information of $T_B^{d,k}(P_d^j)$ is presented in time slots, following the discrete time model discussed in Section 3.2. At this point, the algorithm provides the information of all possible drone options to execute the services individually. The total number of pairs of drone-services ($TotalDronSer$) is given by

$$TotalDronServ = NS \cdot ND \tag{4}$$

In the proposed example, with $NS = 2$ and $ND = 4$ the $TotalDronServ = 8$ pairs of drones-services, from $PairDronServ$ 1 to $PairDronServ$ 8, see Table 4(b). Therefore, in this table are represented all possible service lifetime values depending on the drones to be used. For instance, if S_1 would be executed by drone D_4 ($PairDronServ$ 7), the service lifetime would be $T_B^{d,4}(P_d^1) = 1$ [time slot]; instead, if S_1 would be run on drone D_1 ($PairDronServ$ 1), the service lifetime would be several times greater and equal to $T_B^{d,1}(P_d^1) = 4$ [time slots].

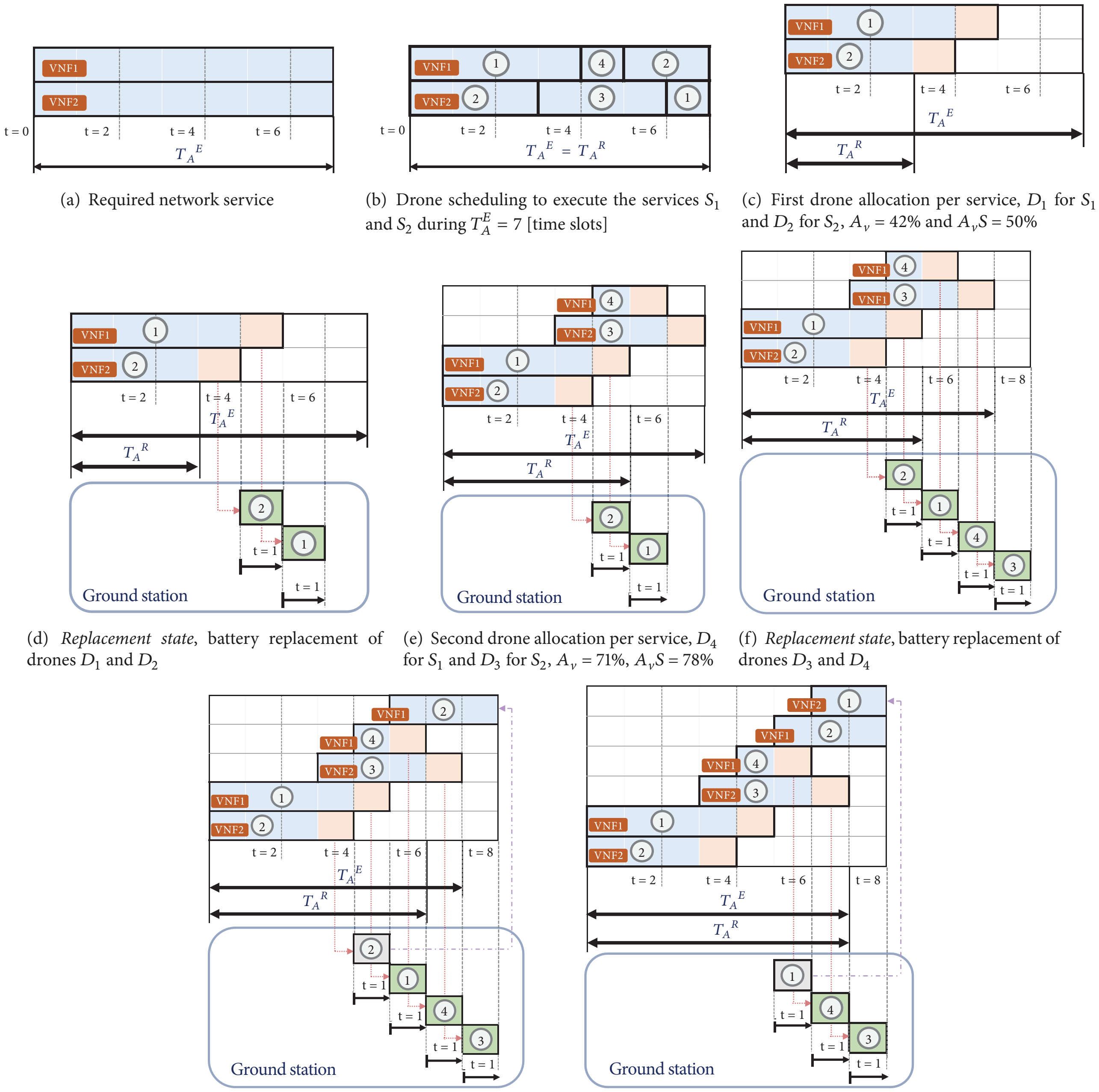
The information of each $PairDronServ$ will be used in the next step of the algorithm to analyze the joint action of drones to run services, in order to obtain the maximum possible A_v and A_vS values in every allocation process.

(4) Combination of drones to run the services: since all services run simultaneously, the different combinations of drones and services ($CombDronServ$) must be analyzed. Hence, the algorithm from a group of ND drones must obtain a set of all possible combinations of drones to run NS services ($AllCombDronServ$). In this context, the total number of combinations ($NumAllComb$) to be processed is given by the analysis of $NS \cdot ND$ pairs ($TotalDronSer$, see Table 4(b)) taken NS at a time and can be expressed as

$$\begin{aligned} NumAllComb &= \binom{ND \cdot NS}{NS} \\ &= \frac{(NS \cdot ND)!}{NS! \cdot (NS \cdot ND - NS)!} \end{aligned} \tag{5}$$

The $NumAllComb$ obtained in this step is critical, because it contributes largely to the growth of complexity of the problem. For instance, $NS = 8$ and $ND = 10$ produce over 28 billion of combinations to be processed.

In accordance with the criteria adopted for the drone scheduling strategy, see Section 3.5, not all $AllCombDronServ$ are valid. For instance, in the example



(a) Required network service

(b) Drone scheduling to execute the services S_1 and S_2 during $T_A^E = 7$ [time slots]

(c) First drone allocation per service, D_1 for S_1 and D_2 for S_2, $A_v = 42\%$ and $A_vS = 50\%$

(d) *Replacement state*, battery replacement of drones D_1 and D_2

(e) Second drone allocation per service, D_4 for S_1 and D_3 for S_2, $A_v = 71\%$, $A_vS = 78\%$

(f) *Replacement state*, battery replacement of drones D_3 and D_4

(g) Third drone allocation per service, D_2 for S_1, $A_v = 85\%$ and $A_vS = 92\%$

(h) Fourth drone allocation per service, D_1 for S_2, $A_v = 100\%$ and $A_vS = 100\%$

Figure 5: Example of the drone scheduling algorithm.

(see Table 4(b)), the combination of *PairDronServ* 1 (D_1, S_1) and *PairDronServ* 2 (D_1, S_2) is not valid, because the same drone (D_1) is assigned to different services (S_1, S_2) at the same time. Likewise, the combination of *PairDronServ* 3 (D_2, S_1) and *PairDronServ* 5 (D_3, S_1) is also not valid, because the same service (S_1) is executed by two drones (D_2, D_3) concurrently. In contrast, the combination of *PairDronServ* 1 (D_1, S_1) and *PairDronServ* 4 (D_2, S_2) is valid, because once selected the drones D_1 and D_2 can be allocated to the services S_1 and S_2, respectively. Considering this, the total number of valid combinations (*NumValComb*) is given by

$$NumValComb = \frac{ND!}{(ND - NS)!} \qquad (6)$$

In the example, $NS = 2$ and $ND = 4$ produce $NumAllComb = 28$ combinations and $NumValComb = 12$ combinations, the latter shown in Table 4(c).

(5) Computation of A_v and A_vS per CombDronServ: using the information of *ValCombDronServ* in Table 4(c) and

Table 2: Parameters related to the processing of combinations and drones in the scheduling algorithm.

Parameter	Description
PairDronServ	Pair of a drone and a service. A pair is used to describe the individual analysis of the execution of a service S_j running on a drone D_k
TotalDronServ	Total number of pairs of drones and services
CombDronServ	Combination of a set of drones to run a set of services. A combination of drones and services, which is commonly referred to as a combination, is composed of different pairs of drones and services
AllCombDronServ	Set of all possible combinations of drones and services
NumAllComb	Total number of all possible combinations. This number is given by (5)
ValCombDronServ	Set of valid combinations of drones and services. The *ValCombDronServ* is a subset of *AllCombDronServ*
NumValComb	Total number of valid combinations. This number is given by (6)
SortCombDronServIDs	Sorted list of the identifiers of the analyzed combinations. To obtain this list, the combinations are sorted in descending order according to the A_v reached
ReplaceDrones	Set of drones whose battery must be replaced
AvDrones	Set of drones that have neither been used nor allocated in the system
ReadyDrones	Set of drones whose battery has been replaced. These drones can be used for a new allocation process
TotAvaDrones	Total number of drones that are available in the system. This number is given by (10)

Table 3: Information of services and drones.

(a) Information of services

S_j	T^j_{init} [*Time slots*]	P^j_d [*Power units*]
1	0	1
2	0	1

(b) Information of drones

D_k	C^k_B [*Power x time slots*]	$T^{r,k}_B$ [*Time slots*]
1	4	2
1	3	2
3	3	2
4	1	2

the $T^{d,k}_B(P^j_d)$ value, given in terms of time slots, in Table 4(b), it is possible to compute the A_v and A_vS metrics, for each of the *CombDronServ*. Table 5(a) shows the different A_v and A_vS values for each *CombDronServ* of Table 4(c). In this table, the metrics have been rounded to the lower bound. An example of the computation of these metrics for the *CombDronServ* 1 is provided below. In the case of *CombDronServ* 1 (composed by *PairDronServ* 1 and *PairDronServ* 4), D_1 is allocated to S_1 during $T^{d,1}_B(P^1_d) = 4$ [time slots], while D_2 is allocated to S_2 during $T^{d,2}_B(P^2_d) = 3$ [time slots].

$$A_{v\,Comb1} = \frac{3\,[time\ slots]}{7\,[time\ slots]} \cdot 100\% = 42.86\% \tag{7}$$

$$\begin{aligned} &A_vS_{Comb1} \\ &= \frac{4\,[time\ slots]\,/7\,[time\ slots] + 3\,[time\ slots]\,/7\,[time\ slots]}{2} \\ &\quad \cdot 100\% = 50\% \end{aligned} \tag{8}$$

(6) Selection of the best CombDronServ: the *ValCombDronServ* are sorted in descending order according to their A_v value (see Table 5(a)), using a quick sort method. The algorithm provides a list with these values and the combination with the best value; the first in the list is selected. Even if there is more than one better combination, the first one in the list is always selected. In the example, the sorted list of all *ValCombDronServ* is

$$\begin{gathered} Sort\ CombDronServ\ IDs \\ \coloneqq \{1, 2, 4, 5, 7, 9, 3, 6, 8, 10, 11, 12\} \end{gathered} \tag{9}$$

where the *CombDronServ* 1 ($ID = 1$) is the best combination, while the *CombDronServ* 12 has the lowest A_v level. After selecting the best *CombDronServ*, in the example *CombDronServ* 1, the algorithm proceeds to identify the drones and services belonging to that combination, as shown in Table 5(b).

(7) Drone allocation and computation of T^R_A, A_v, and A_vS: in this step, with the information of the best combination, the algorithm allocates the drones to their corresponding services over time. As shown in Table 6(a), D_1 is allocated to S_1 and D_2 is allocated to S_2. Figure 5(c) illustrates this allocation procedure, and in this figure is not only represented the

Table 4: Information of drones, services, and combinations.

(a) Information about battery lifetime of drones for services S_1 and S_2

D_k	$T_B^{d,k}(P_d^1)$ $[Time\ slots]$	$T_B^{d,k}(P_d^2)$ $[Time\ slots]$
1	4	4
2	3	3
3	3	3
4	1	1

(b) Information of pairs of drones and services and drone lifetime per time slot

No.Pair	D_k	S_j	*Drone per time slot*
1	1	1	1 1 1 1
2	1	2	1 1 1 1
3	2	1	2 2 2 0
4	2	2	2 2 2 0
5	3	1	3 3 3 0
6	3	2	3 3 3 0
7	4	1	4 0 0 0
8	4	2	4 0 0 0
		Time Slot	1 2 3 4

(c) Combinations among available drones to run the services

No.Comb.	*Combination* (*No.Pair*)	*Drones*	*Services*
1	1, 4	1, 2	1, 2
2	1, 6	1, 3	1, 2
3	1, 8	1, 4	1, 2
4	2, 3	1, 2	2, 1
5	2, 5	1, 3	2, 1
6	2, 7	1, 4	2, 1
7	3, 6	2, 3	1, 2
8	3, 8	2, 4	1, 2
9	4, 5	2, 3	2, 1
10	4, 7	2, 4	2, 1
11	5, 8	3, 4	1, 2
12	6, 7	3, 4	2, 1

Table 5: Computation of metrics for *ValCombDronServ* and selection of the best combination.

(a) Computation of A_v and A_v for all *ValCombDronServ*

No.Comb.	*Combination* (*No.Pair*)	A_v(%)	A_vS(%)
1	1, 4	42	50
2	1, 6	42	50
3	1, 8	14	35
4	2, 3	42	50
5	2, 5	42	50
6	2, 7	14	35
7	3, 6	42	42
8	3, 8	14	28
9	4, 5	42	42
10	4, 7	14	28
11	5, 8	14	28
12	6, 7	14	28

(b) Information of pairs of drones and services belonging to the best combination

No.Pair	D_k	S_j	*Drone per time slot*
1	1	1	1 1 1 1
4	2	2	2 2 2 0
		Time Slot	1 2 3 4

execution state ($T_B^{d,k}(P_d^j)$, blue color) but also the *replacement state* ($T_B^{r,k}$, orange and green colors). Then, the performance metrics A_v and A_vS are computed. In this particular example the values reached are $A_v = 42\%$ and $A_vS = 50\%$. This is the initial allocation of drones, and since none of the stop criteria have been met, i.e., $A_v \neq 100\%$ and the system has available resources (drones D_3 and D_4), the algorithm continues its execution process.

(8) Identification of drones to replace their battery (ReplaceDrones): the algorithm must constantly monitor the drones used, so that when they finish their execution (T_A^R), they have to change to the *replacement state* (*ReplaceDrones*). In the example, in the first allocation drones D_1 and D_2 have been used, and, as can be seen in Figure 5(d), drone D_2 must start its replacement process in time slot 4; instead, D_1 must perform this procedure in time slot 5.

(9) Identification of services to be executed by the available drones (AvDrones) in the next allocation process: to continue with the drone scheduling process, from the first allocation, an analysis of the priority of the executed services is carried out. This level of priority is given based on the (T_A^R) parameter of the services. Thus, a service with a lower (T_A^R) value will have a higher priority level to be processed in the next drone allocation step. In this way, the algorithm will allocate the drones with the highest ($T_B^{d,k}(P_d^j)$) values to the services with the highest priority levels. The priority in the execution time of the services is analyzed at the end of the first allocation process, since the start time of all the services is the same ($T_{init}^j = 0$).

The priority information of the services is used in the computation of the combinations and in the selection of the best combination, from the second allocation. This process is carried out to guarantee an efficient and uniform allocation of the drones; otherwise, a specific service could achieve a T_A^R value much higher than the others. This situation must be avoided since the services must be executed simultaneously with the objective of always reaching an A_v as large as possible. In the example (see Figure 5(d)), S_2 ($T_A^R = 3$ [time slots]) has higher priority level compared to S_1 ($T_A^R = 4$ [time slots]). Therefore, later in the computation of all combinations this information will be considered so that the selection of the best combination allows a drone allocation ($AvDrones = 2$, $D3$ and $D4$) that favors the execution of S_2, as shown in Figure 5(e).

The priority level of services is also useful in the case that $ND < NS$, a situation that may occur after the first drone allocation. In this particular case, the priority allows to know the services that must be attended, with the available resources. In this scenario, the combinations are computed with the number of available drones.

(10) Identification of total available drones (TotAvaDrones): the total number of drones available in the system include drones that have not been used, which in the case of the example are D_3 and D_4(*AvDrones*), and drones whose battery has been replaced and are ready (*ReadyDrones*) to be used when the system demands them. The *ReadyDrones* are the *ReplaceDrones* that have completed the *replacement state*. In the example proposed, at this point, the first drone allocation has been performed; therefore, the system does not have *ReadyDrones*. Under these conditions, the drones available are D_3 and D_4. In the next stages, the algorithm could have *ReadyDrones* available, once they exceed the *replacement state*.

The total number of available drones (*TotAvaDrones*) in the system can be expressed as

$$TotAvaDrones = AvDrones + ReadyDrones \qquad (10)$$

(11) Iterative drone allocation: in an iterative process of computation of combinations, selection of the best combination, drone allocation, priorities of services and metrics, the algorithm continues its execution until either of the two stopping criteria is met (A_v = 100% or $TotAvaDrones$ = 0). Continuing with the example, in the second iteration the algorithm allocates D_3 to S_2 and D_4 to S_1, as shown in Table 6(b) and in Figure 5(e). The results after this procedure are A_v = 71% and A_vS = 78%. In a similar way to the process carried out at the end of the first iteration, once the services have been completely executed by drones D_3 and D_4, these devices pass to the *replacement state*. Specifically, this status will be reached at time slot 5 for D_4 and at time slot 6 for D_3, as represented in Figure 5(f). For practical reasons, the time window in proposed example has been limited to 8 [time slots].

After the second iteration has been completed, the algorithm checks the total number of available drones. At this point, given that $AvDrones = 0$, the algorithm checks if any of *ReplaceDrones* has become *ReadyDrones*. In the example, the unique drone that meets this condition is D_2, which after the corresponding computations is allocated to S_1, as shown in Table 6(c) and in Figure 5(g). At the end of the third iteration A_v = 85% and A_vS = 92%. Once the third iteration is finished, the scheduling process continues in the same way as in the previous iterations and according to the steps described in the Figure 4. The fourth iteration is the last in the example. In this iteration $D1$, whose battery has been replaced, is allocated to S_2, as seen in Figure 5(h) and in Table 6(d). Resulting in the final values: $T_A^R = T_A^E = 7$ [time slots] for both services (S_1 and S_2), A_v = 100% and A_vS = 100%. Once these values are obtained, the algorithm stops its execution.

A summary of the complete drone allocation procedure is depicted in Figure 5(b). The final drones sequence is D_1, D_4, and D_2 for S_1 and D_2, D_3, and D_1 for S_2. Moreover, as a relevant result the algorithm allows knowing the number of drones (resources) needed to reach a certain availability level. In the example, during a time window of $T_A^E = 7$ [time slots], with 4 drones and an optimal scheduling, the system can reach an A_v = 100% in the services execution.

Table 6: Progressive allocation of drones to fulfill the network services.

(a) Initial drone allocation

S_j	*Drone information per time slot*
1	1 1 1 1 0 0 0
2	2 2 2 0 0 0 0
Time Slot	1 2 3 4 5 6 7

(b) Second drone allocation

S_j	*Drone information per time slot*
1	1 1 1 1 4 0 0
2	2 2 2 3 3 3 0
Time Slot	1 2 3 4 5 6 7

(c) Third drone allocation

S_j	*Drone information per time slot*
1	1 1 1 1 4 2 2
2	2 2 2 3 3 3 0
Time Slot	1 2 3 4 5 6 7

(d) Final drone allocation

S_j	*Drone information per time slot*
1	1 1 1 1 4 2 2
2	2 2 2 3 3 3 1
Time Slot	1 2 3 4 5 6 7

4.2. Complexity Analysis. The NS and ND values have an impact on the growth of complexity of the algorithm. The growth rate of the problem is not linear, because it depends on the product of $NS \cdot ND$, as shown in (4) and in Table 4(b).

The steps 3 and 4 define the growth of the algorithm. This growth rate as a function of NS and ND can be expressed as

$$f(NS \cdot ND) = NS \cdot ND + C(NS \cdot ND, NS) \qquad (11)$$

where the second term is the dominant term within the expression. As described in Section 4 it represents total set of all combinations (*AllCombDronServ*) that the algorithm must analyze in a mandatory manner in order to find the valid combinations *ValCombDronServ* to be processed.

Thus, according the Big-O classification [16], ignoring the low-order terms, i.e., the first term in (11), the order of growth of the drone scheduling algorithm is $\mathcal{O}(C(NS \cdot ND, NS))$. Hence, this complexity reveals the drawbacks that the algorithm has for the selection of NS and ND values.

5. Performance Evaluation

To validate the resource planning algorithm proposed in the previous section, it is necessary to define reasonable scenarios that can integrate all the different parameters that should be assessed and provide the complex environment where this type of algorithms is normally applied. The evaluation will then be carried out through extensive simulations using these scenarios. Section 5.1 describes the simulation setup and the two application scenarios and then the evaluation results are presented and discussed in Section 5.2.

5.1. Simulation Setup. The drone scheduling strategy is evaluated in two different application scenarios: a *Generic Scenario* that uses random values for some parameters and a *Realistic Scenario* whose values are based on experiments and measurements, as shown in Table 7. These scenarios are described in detail below, and, for practical reasons, in both scenarios the NS has been limited up to $NS = 7$ services. The scheduling algorithm has been implemented using Matlab (Matlab R2017b).

The *Generic Scenario* has been run on a computer equipped with a 3.33 GHz x 12 cores Intel Core i7 Extreme processor and 12 GB RAM. This simulation leverages parallel processing and multiple CPU cores (up to 6 cores) have been used in each simulation. In order to ensure stability of the results, each case (NS) was repeated 50 times except for $NS = 7$, which was executed only once due to its excessive running time. Results are shown, in Figures 10 and 11, with a confidence interval of 95%. The total running time for this scenario (all cases) exceeded 300 hours.

Meanwhile, the *Realistic Scenario* has been executed on a machine with a 3GHz x 4 cores Intel Core i5-7400 processor and 64 GB RAM. In this case parallel processing has also been exploited, and all 4 cores have been used during the simulation. Because this scenario is deterministic, only one simulation has been executed for each case ($T_A^E = 5$, $T_A^E = 10$ and $T_A^E = 15$ hours). The execution time for this scenario is around 80 hours, and results are shown in Figure 12.

5.1.1. Generic Scenario. This scenario corresponds to a very general application environment with the intention of performing an initial validation of the proposed solution. To this end, it is assumed that the different drones can execute the services in the air (with a much higher power consumption), a subset can land on the ground after its launch from the ground control station, or even a hybrid situation can also be possible (different cases are discussed in Section 3.1).

The scenario is not particularized for specific applications and it is considered that applications can vary from the provision of video surveillance services to the provision of connectivity services, etc. (this is modeled in the scenario by considering the power demanded by the different services P_d^j to be a random value between 1 and 5 A). In addition, to provide more diversity, batteries capacities C_B^k are considered to be different for each drone, choosing for them random values between 1 and 5 Ah. This assumption (see Table 7) will produce services with different T_A^R values (*service execution state*) varying from $T_A^R = 0.2$ [hours] (minimum value) to $T_A^R = 5$ [hours] (maximum value).

Considering the parameters described above, a $T_B^{r,k} = 10$ [minutes] (*replacement state*) and a time window of $(T_A^E) = 10$ [hours], the algorithm has to perform many transitions among the *service execution state* and the *replacement state* in order to optimally allocate the available resources to the corresponding services, which is exactly the situation that is wanted to be forced in this scenario in order to test the algorithm capabilities. The metrics achieved for the different NS and ND values are shown later in Section 5.2.

5.1.2. Realistic Scenario

Scenario Definition. The objective of this scenario is to test the algorithm under more real conditions replacing the random values used in the Generic Scenario by some other values that may be closer to some real ones.

In particular the scenario that will be described in this section (Figure 6) shows a set of drones each one with an onboard SBC (they carry an RPi with its own battery) linked through an ad hoc WiFi network and using a certain FANET (Flying Adhoc Network) routing protocol to guarantee connectivity. The drones are including different VNFs depending on the role they are assuming (Access Point (AP), router, or telemetry transmitter (i.e., video or sensor data).

In this scenario the energy consumption for a particular drone may depend on many diverse factors. In first place there are two different types of batteries and also drones that are flying and drones that are landed (so depending on the situation the battery that limits the service maybe either one or the other). For the drones that are not flying (the drone battery is not presenting any limitations for them and only the RPi battery is used) the measurements must consider the different WiFi interfaces, the WiFi communications (different traffic including video, telemetry, routing messages, etc.), CPU load, external hardware, etc.

As it can be appreciated it is not easy in this environment to evaluate how the energy consumption curve will perform for the different drones and how many drones are in fact needed in order to guarantee that the service can be maintained over time (considering a certain replacement time), etc. This is considered to be a suitable scenario so as to validate the combinatory algorithm and the rest of the section will provide more details on the scenario itself and about the validation methodology.

A total of seven drones have been considered in this scenario that may represent a natural disaster use case (e.g., earthquake, fire, flood) where drones can enable communications between emergency services islands; as seen in Figure 6 drones accommodate different VNFs and play different roles within the network to perform the overall service that will be taken into account by the algorithm:

(1) *Optimized Link State Routing (OLSR) Router VNF.* Because of the drone network nature (e.g., node mobile, volatile network) OLSR [17] has been selected as routing protocol. OLSR is a distributed and proactive routing protocol used to establish connections between participant nodes in an ad hoc wireless network proposed for Mobile Ad hoc Networks (MANETs) and extended for Flying Ad hoc Networks (FANETs). The main advantage of this type of routing protocol is its dynamic discovery allowing stateless

Table 7: Summary of simulation parameters.

Scenario	T_A^E	NS	ND	T_{init}^j	C_B^k	P_d^j	$T_B^{r,k}$
Generic	10 hours	1-7	0-11	0, for all j	uniform distributed random value [1- 5] [Ah], for all k	uniform distributed, random value [1- 5] [A], for all j	10 [min], for all k
Realistic	5, 10, 15 hours	7	0-10	0, for all j	3 [Ah], for all k	S1 Router: 292.02 [mA] S2 Router: 292.35 [mA] S3 AP + Router: 371.82 [mA] S4 AP + Router: 373.62 [mA] S5 Telemetry TX: 288.76 [mA] S6 Telemetry TX: 288.23 [mA] S7 Flying: 9000 [mA]	10 [min], for all k

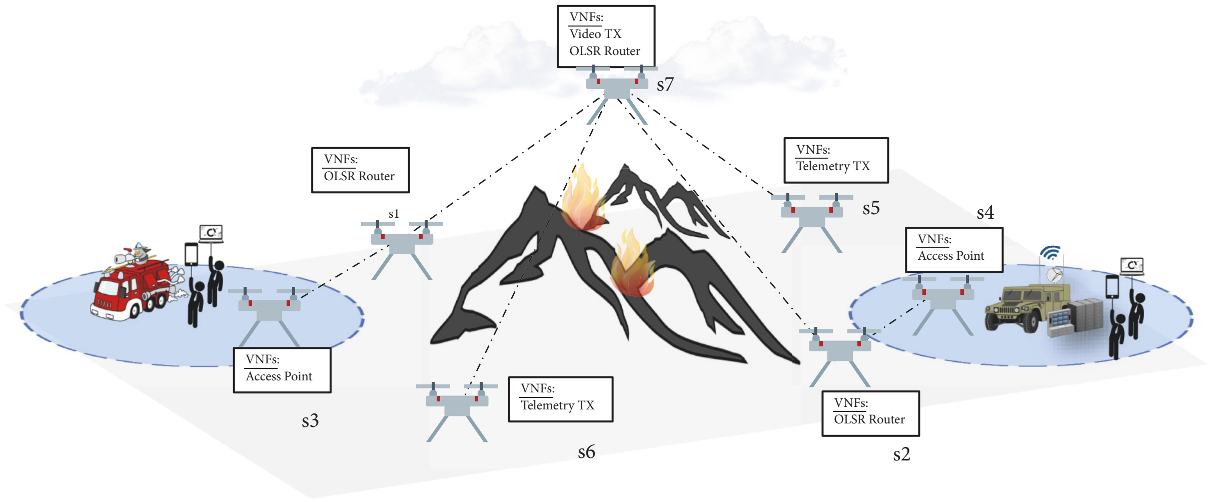


Figure 6: Drone swarm providing network connectivity in a disaster situation.

VNFs that prevent the costly process of VNF migration. Drones s1 and s2 in Figure 6 are OLSR routers.

(2) *Access Point VNF.* Wireless AP VNFs are used to interconnect wireless communication equipment from emergency services (end user terminals). The selected technology has been the normal 2.4 GHz IEEE 802.11. Drones s3 and s4 in Figure 6 are APs.

(3) *Telemetry Transmitter VNF.* Data transmission VNFs have been used in different drones within the network. As it can be appreciated in Figure 6, two types of data transmitters have been specified, (a) telemetry transmitter (32 Kbps flow that can either represent GPS information or sensor data such as temperature or humidity) and (b) video transmitter in standard quality (200 Kbps flow) that is enough to have an overview of the disaster area by the emergency services. Drones s5 and s6 in Figure 6 are telemetry transmitters.

Drones s1 to s6 are landed and the energy demanded is only related to the network services while drone s7 is flying (Figure 6).

Parameters Estimation. In order to validate the algorithm, it is necessary to provide as input a rough estimation of the power demanded by each drone. However, in this realistic environment, besides user traffic (video, telemetry, etc.), numerous factors may affect battery lifetimes. Parameters like the pattern of energy consumption, environmental conditions, or battery status are significant factors to take into account in real applications, but it is extremely complex to model them in simulated environments therefore despised. However, there is unpredictable network traffic which is considered to measure energy consumption, such as packets retransmission, WiFi management packets, routing messages, etc.

The power consumption will be directly measured using a real RPi and a specific power meter. In order to do so, it is required that the RPi resembles the real conditions as stated in the scenario definition in terms of traffic and CPU load and considers the necessary hardware to enable wireless communication since the consumption depends heavily on these parameters.

To calculate all these values, a simulation using ns-3 (ns-3: https://www.nsnam.org/) network simulator has been performed. As it can be seen in Figure 6, the simulation

Table 8: Simulation details.

Parameter	Values
Traffic	CBR
Telemetry Transmission Rate	32 Kbps
Video Transmission Rate	200 Kbps
Network Protocol	UDP
Routing Protocol	OLSR
Simulation Time	3600 seconds
Number of Drones	7
Mobility Model	Static

includes the seven drones (one of them is flying (s7) while the rest of them are landed). The drones that accommodate the telemetry transmitter VNFs generate one flow to each one emergency service involved on the scene. More details can be consulted in Table 8 where the simulation parameters are specified. The trajectory of the flying node (s7) has been precalculated using Matlab and included in the simulation using traces with the ns format.

After this simulation, it will be possible to calculate the traffic that will be processed by each drone, including all the different components that have been previously mentioned.

To be able to properly analyze the traffic at each drone, the following characterization has been done for the traffic depending on the source and the destination as represented in Figure 8:

(1) *Transit traffic:* received traffic to be forwarded because that drone is not the final destination of the packet. This traffic is telemetry data.

(2) *Sink traffic:* traffic that is consumed by that particular drone. This traffic corresponds to OLSR management or WiFi management packets since telemetry data is consumed out of the analyzed network by the emergency services.

(3) *Source traffic:* traffic that is generated at that particular drone. This traffic can be either OLSR management, WiFi management, or telemetry data.

Figure 7 shows the average throughput for each drone obtained by the simulation. These results will be replicated into real RPis in order to perform the power measurements. Note that the flying node s7 transmitting video is not represented in the figure. Flight-engines are demanding all the available energy in this drone and power consumption due to traffic processing is negligible in comparison. No power measurement is performed here since the battery that limits the service execution is the one of the drone itself (in Table 7 this power consumption is modeled as 9000 [mA] since together with the battery capacity that is used, a 20 minutes flight estimation is obtained which is quite normal for regular drones). In addition, as expected the traffic consumed by drone (only OLSR management) is insignificant compared with transit and generated traffic.

Power Consumption Measurements. As it has been mentioned, the main purpose of the simulations described in the previous section is to estimate the different data flows (Figure 7) that have to be injected into the RPi in order to emulate the power consumption that it would have in a real scenario.

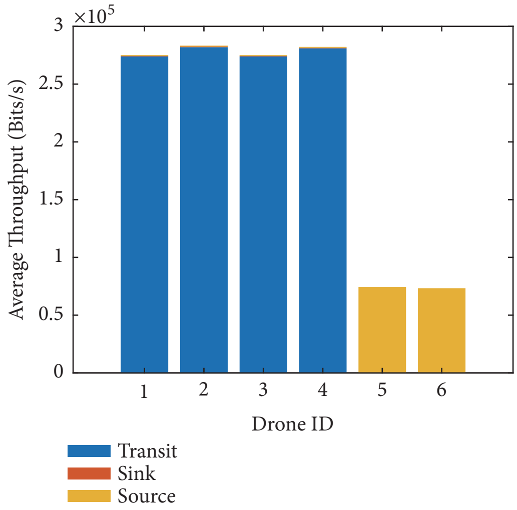


Figure 7: Average throughput for Realistic Scenario.

In order to perform these measurements, the testbed depicted in Figure 8 has been built. It includes three different RPis 3B and the Monsoon FTA22D Power Meter (Monsoon FTA22D Power Meter: https://www.msoon.com/) (which provides a robust power measurement solution for mobile devices with high accuracy (±1%)).

The RPi source generates two flows, one of them consumed by the RPi hosting the VNF and the second one consumed by the RPi destination. The RPi that accommodates the VNF is also generating another flow that is consumed by the RPi destination. In this way, we can emulate the traffic involved with each device on the network, to generate this traffic, the Iperf (Iperf: https://iperf.fr/) tool.

The RPi VNF is then powered by the Monsoon (Vout voltage of 4.2 V) main channel and then the average power is derived from instantaneous current (Figure 9) and voltage and divided by the duration of the sampling run (200 seconds). After conducting the measurements, it has been verified that the power consumption is not significantly increased unless the network interface is close to the maximum bandwidth. The authors in [8] are finding similar results. The most relevant power increase is due to the use of an external hardware (extra WiFi card to create the AP) carried by drones s3 and s4.

5.2. Results. In both scenarios, *Generic* and *Realistic*, the optimal drone scheduling is performed. As a result, the algorithm allows knowing the level of services availability achieved when using a given number of available drones. Seen in another way, the proposed strategy can be used to know how many drones need to be deployed to reach a certain services availability level (in the simulations from $A_v = 0\%$ to $A_v = 100\%$). In this context, the results provided by the drone scheduling algorithm can be used in the design, planning, and deployment stages of network services

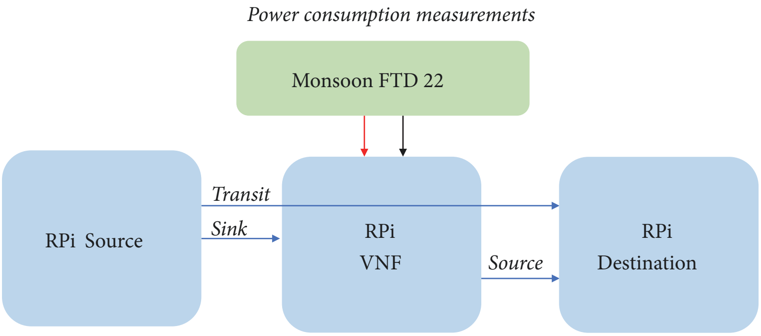


Figure 8: Power consumption measurements methodology.

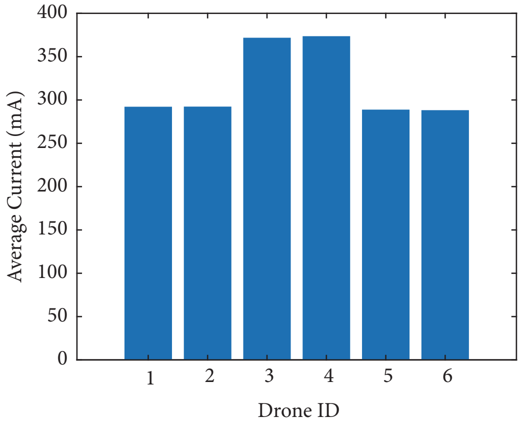


Figure 9: Average Current for Realistic Scenario.

executed by drones. Therefore, the information provided by the scheduling strategy can be used for both performance evaluation and system dimensioning.

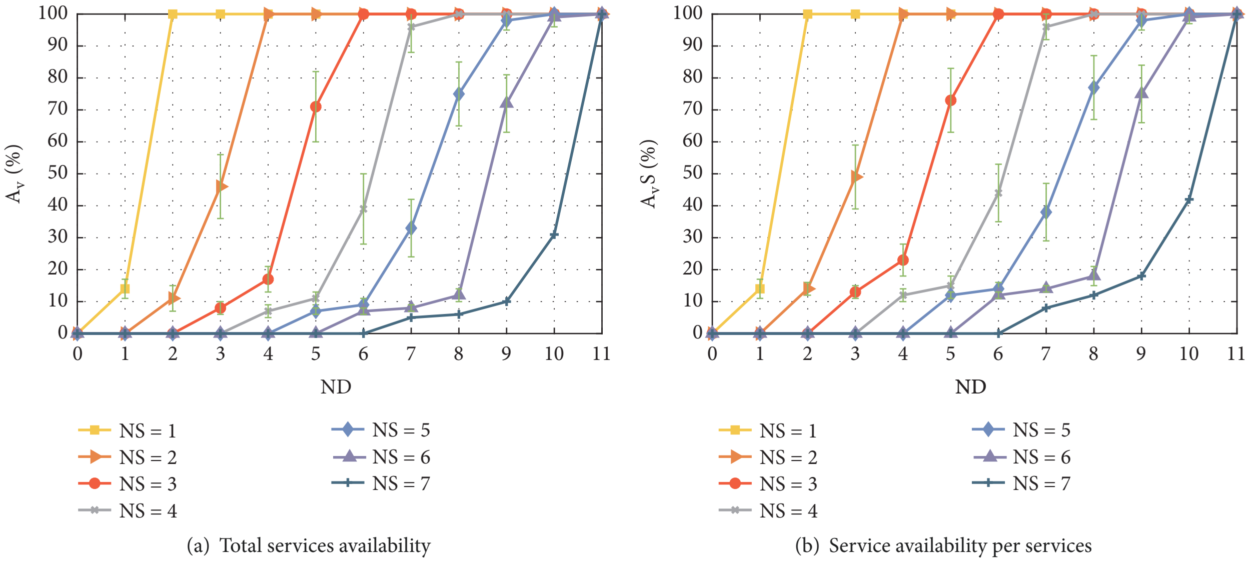
Regarding the *Generic Scenario*, whose results are shown in Figure 10(a) (A_v), in Figure 10(b) (A_vS), and in Figure 11 ($A_v = 100\%$), the algorithm provides the different availability levels (metrics) for each of the services from $NS = 1$ up to $NS = 7$. As discussed above, this information allows knowing the number of drones needed to reach a certain A_v value, or the A_v obtained with a given amount of available resources (drones). For example, as shown in Figure 10(a), with $NS = 3$ services are needed $ND = 6$ drones to reach an $A_v = 100\%$ (also $A_vS = 100\%$ in this case). Similarly, if as system consists of $NS = 5$ services and demand $A_v = 90\%$, the number of drones required is ND $= 9$ drones. In the latter example, as shown in Figure 10(a), for $NS = 5$ there is no an exact ND value that meets $A_v = 90\%$; in this case the selection should round the upper bound value ($ND = 9$), due to the fact that the ND is an integer number. This rounding operation also ensures that the value obtained is greater than or equal to the requested value.

Another relevant result that can be extracted from the results provided by the algorithm (Figure 10(a)) is ND as a function of NS to reach $A_v = 100\%$, as shown in Figure 11 (also $A_vS = 100\%$). This summarized information represents a practical and useful tool that can be used in the design and planning phases of services that have as a constraint the 100% of availability for their operation (e.g., provision of communications services in emergency or search scenarios). For example, in the *Generic Scenario*, it is appreciated that for $NS = 6$ are needed at least $ND = 10$ drones to reach and $A_v = 100\%$.

In addition, the results obtained help corroborate the criteria that were considered in the design of the algorithm. For example, as discussed in Section 3.4, all A_vS values must be equal or greater than A_v values. This condition is verified by establishing a comparison between Figure 10(a) (A_v) and Figure 10(b) (A_vS) for the *Generic Scenario*, and between Figure 12(a) (A_v) and Figure 12(b) (A_vS) for the *Realistic Scenario*. For example, for the (*Generic Scenario*), with $NS =$ 5 services and $ND = 7$ services, $A_v = 33\%$ instead of $A_vS = 38\%$.

In the *Generic Scenario* in specific, the A_v and A_vS metrics obtained are very similar to each other (always $A_v \geq A_vS$). This situation obeys one or more of the following considerations: (i) T_A^E and T_A^R are large compared to the size of time slots (minimum amount of time in the system, 10 [min] in the simulations) and (ii) in the final allocations there is not much difference between T_A^R of all services (i.e., C_B^k) and (P_d^j) have similar values for all services). In the *Realistic Scenario* instead, A_v (Figure 12(a)) and A_vS (Figure 12(b)) values are different from each other, mainly for $T_A^E = 5$ [hours] and $T_A^E = 10$ [hours]; this due to the considerable difference of T_A^R for all services, in particular referred to S_7, whose demanded consumption (9 [A]) is much greater than the rest of services. In this case, the algorithm needs to allocate a greater number of drones to execute the service, or in other words to keep the service in active mode a high replacement rate is necessary (i.e., high transition between the *service execution state* and the *replacement state*).

On the other hand, in Figure 12(a) (A_v) and Figure 10(b) (A_vS) are shown the performance metrics achieved for the *Realistic Scenario*. In this scenario, all cases ($T_A^E = 5$, 10 and 15 [hours]) consider $NS = 7$ services, and as a result of the evaluation, the algorithm provides that the required ND to reach $A_v = 100\%$, in all cases, is equal to $ND =$ 10 drones. This value represents the minimum amount of resources (drones and batteries) that the system must use to face a reliable ($A_v = 100\%$) network services deployment.



(a) Total services availability

(b) Service availability per services

FIGURE 10: Performance evaluation of the drone scheduling strategy, Generic Scenario.

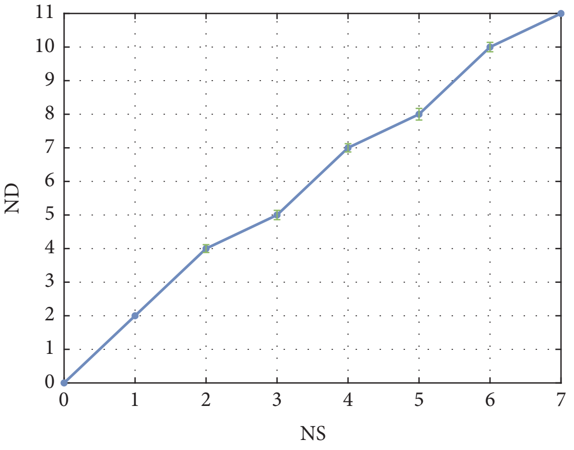


FIGURE 11: *ND* to reach an $A_v = 100\%$ - Generic Scenario.

In this scenario, different T_A^E values have been considered, to evaluate the behavior of the strategy in both short-term and long-term real applications. In this regard, the scheduling algorithm performs a few numbers of allocation procedures when $T_A^E = 5$ [hours] and $T_A^E = 10$ [hours] (only S_5 needs to used different resources); instead, a high transition between *service execution* and *state* is experienced when $T_A^E = 15$ [hours].

6. Conclusions

In this paper, an optimal drone scheduling algorithm is developed, which by leveraging 5G and NFV capabilities is able to perform an efficient energy-aware management of resources for network services provisioning. Through this strategy, it is possible to calculate the required number of drones for a certain degree of service, to be used in real scenarios. The scheduling strategy, based on two states, *service execution* and *replacement*, provides the information about the number of drones and their sequence of replacement to run services and reach a certain availability level, during a finite time interval. Thus, the proposed scheduling algorithm can be used as a useful tool in system dimensioning and missions planning tasks, in order to provide reliable and safe drone-based network services deployments.

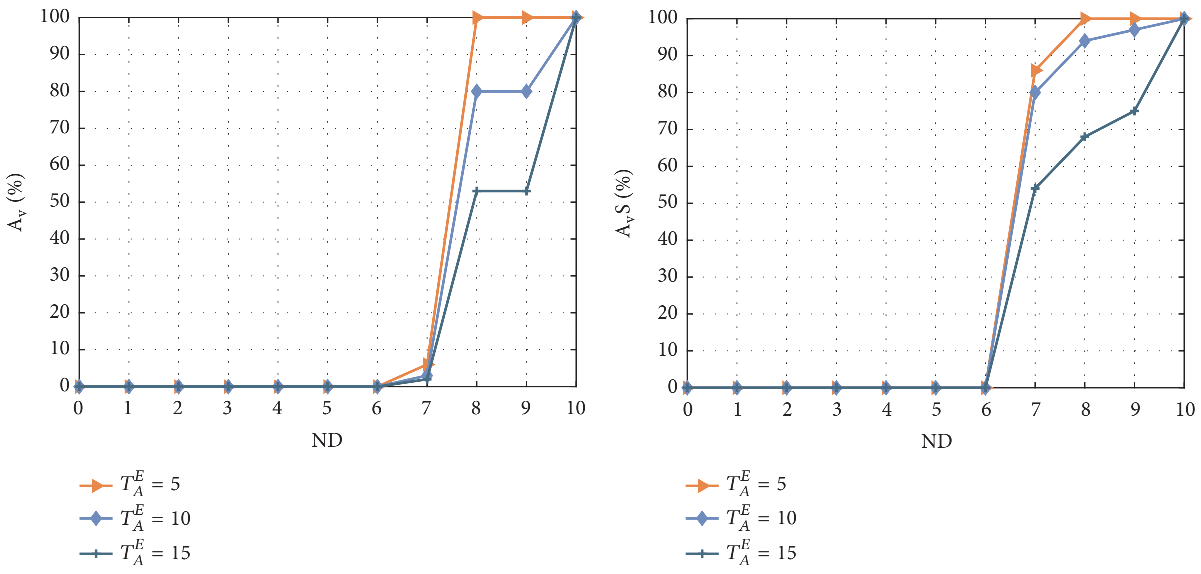
The algorithm can perform the optimal scheduling in both short- and long-term applications, and it can be used as a resource/availability planner in a wide variety of real scenarios, such as emergency scenarios, relief disaster services, and search and rescue tasks.

Simulations results validate the performance of the proposal and provide the metrics achieved, as well as the amount of resources needed for the execution of services in different scenarios.

The results provided by the simulations can be used to know the level of availability for a certain number of services and available drones. Likewise, these results allow knowing the number of drones needed to run services to guarantee 100% of availability level.

Finally, the paper presents the evaluation of the proposal for scenarios up to $NS = 7$ services and $ND = 11$ drones, limited by to the complexity of the algorithm. Although these limits can be very useful and quite adequate values in many practical scenarios and applications, it is necessary to develop additional strategies in order to be able to handle larger scenarios and in a faster running time. In this regard, the brute-force search solution developed is also useful as a baseline method to design heuristics or metaheuristics approaches. Currently, authors are working on developing these solutions as part of the future works.

Based on information provided by the implemented strategy, the future works also include the modeling of the services availability as a mathematical function, in terms of



(a) Total services availability

(b) Service availability per services

Figure 12: Performance evaluation of the drone scheduling strategy, Realistic Scenario.

the number of services, their power consumed, the capacity of the batteries, and the number of drones.

Data Availability

The data and results used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been supported by the Ministerio de Economía y Competitividad of the Spanish Government under projects TEC2016-76795-C6-1-R and TEC2016-76795-C6-3-R and also AEI/FEDER, UE. Christian Tipantuña acknowledges the support from Escuela Politécnica Nacional (EPN) and from Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) for his doctoral studies at Universitat Politécnica de Catalunya (UPC).

References

[1] H. Shakhatreh, A. Sawalmeh, A. Al-Fuqaha et al., “Unmanned aerial vehicles: A survey on civil applications and key research challenges,” 2018, https://arxiv.org/abs/1805.00881.

[2] 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 *Proceedings of the Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, pp. 39–44, ACM, 2018.

[3] 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,” in *Proceedings of the Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*, pp. 522–527, IEEE, 2017.

[4] 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, 2018.

[5] M. Sara, I. Jawhar, and M. Nader, “A softwarization architecture for UAVs and WSNs as Part of the cloud environment,” in *Proceedings of the Cloud Engineering Workshop (IC2EW), 2016 IEEE International Conference on*, pp. 13–18, IEEE, 2016.

[6] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in UAV communication networks,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[7] Y. Zeng and R. Zhang, “Energy-efficient uav communication with trajectory optimization,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[8] F. Kaup, P. Gottschling, and D. Hausheer, “PowerPi: measuring and modeling the power consumption of the raspberry Pi,” in *Proceedings of the Local Computer Networks (LCN), IEEE 39th Conference on*, pp. 236–243, 2014.

[9] A Carroll, G. Heiser et al., “An analysis of power consumption in a smartphone,” in *Proceedings of the In USENIX annual technical conference*, vol. 14, p. 21, Boston, Mass, USA, 2010.

[10] P. Serrano, A. Garcia-Saavedra, G. Bianchi, A. Banchs, and A. Azcorra, “Per-frame energy consumption in 802.11 devices and its implication on modeling and design,” *IEEE/ACM Transactions on Networking (ToN)*, vol. 23, no. 4, pp. 1243–1256, 2015.

[11] S. Dunbar, F. Wenzl, C. Hack et al., “Wireless far-field charging of a micro-UAV,” in *Proceedings of the Wireless Power Transfer Conference (WPTC)*, pp. 1–4, IEEE, 2015.

[12] M. Wang, “Systems and methods for uav battery exchange,” 2016, US Patent 9,346,560.

[13] K. A. O. Suzuki, P. Kemper Filho, and J. R. Morrison, “Automatic battery replacement system for UAVs: Analysis and design,” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 563–586, 2012.

[14] C. Tipantuña and X. Hesselbach, "Demand-Response power management strategy using time shifting capabilities," in *Proceedings of the Ninth International Conference on Future Energy Systems*, pp. 480–485, ACM, 2018.

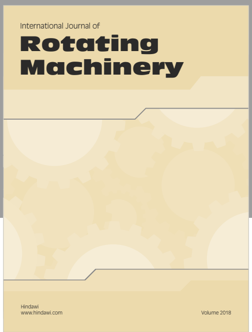
[15] ETSI Industry Specification Group (ISG) NFV, "ETSI GS NFV 002 V1.1.1: Network Functions Virtualisation (NFV); Architectural Framework," Tech. Rep., 2013.

[16] S. Johnson Michael, R. Garey, and David, *Computers and Intractability: A Guide to the Theory of NP-completeness*, WH Free. Co., San Francisco, Calif, USA, 1st edition, 1979.

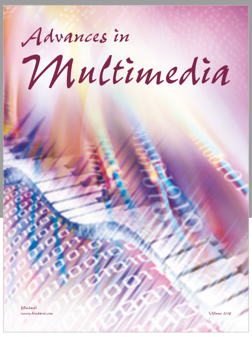
[17] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," Technical report, 2003.

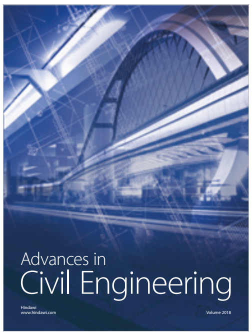
Journal of
Engineering

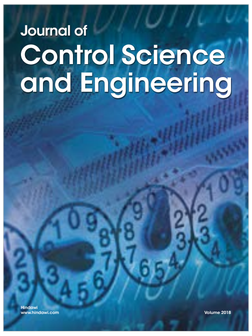
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

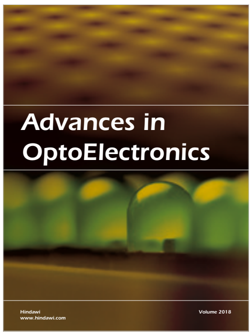

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

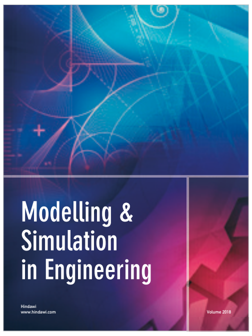
Journal of
Robotics

Journal of
Electrical and Computer Engineering

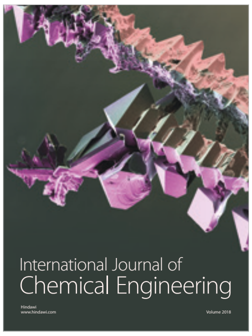

Advances in
OptoElectronics

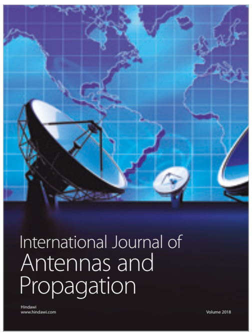
VLSI Design

International Journal of
Navigation and Observation

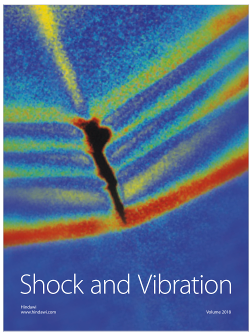

Modelling & Simulation in Engineering

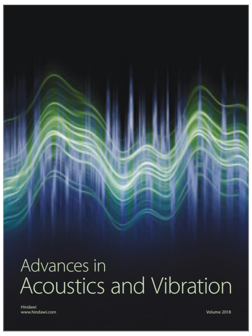
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration