

# Optimizing FANET Lifetime for 5G Softwarized Network Provisioning

G. M. Cappello<sup>1</sup>, G. Colajanni<sup>1</sup>, P. Daniele<sup>1</sup>, L. Galluccio<sup>2</sup>, C. Grasso<sup>2</sup>, G. Schembra<sup>2</sup>, L. Scrimali<sup>1</sup>

<sup>1</sup>Department of Mathematics and Computer Science

<sup>2</sup>Department of Electrical, Electronics and Informatics Engineering (DIEEI) and CNIT Research Unit

University of Catania

Viale Andrea Doria, 6 - 95125 Catania, Italy

**Abstract**—Recently, Flying Ad Hoc Networks (FANET) have been proposed to empower 5G networks to support complex missions and provide ubiquitous connectivity to heterogeneous devices. However, it is needed to cope with the limited UAV capabilities (e.g. limited available energy to supply engines and computing elements, limited computing capabilities), as well as with the need to provide network and application services as foreseen in highly dynamic and time varying 5G ecosystems.

This paper presents for the first time a comprehensive framework that integrates a FANET with a 5G network, with the aim of providing services that can be even chained with each other. This model is comprehensive in the sense that it takes into account physical constraints of the devices, as well as features and requirements of traffic flows. For this framework, the paper proposes a mathematical optimization model, allowing Virtual Function (VF) placement and chaining, aimed at minimizing energy consumption and service unsatisfaction probabilities of the FANET as a whole without employing heuristics for the solution of the problem. Two placement strategies named MLP and WMP are introduced and compared with the standard placement strategy named NoShP. An extensive numerical analysis shows that MLP and WMP allow us to well catch network dynamics and to reduce the number of virtual functions needed while decreasing the power consumption, so increasing UAV flight time and network lifetime.

**Index Terms**—5G, NFV, UAV, Service Chains, Mathematical Optimization, Virtual Function Placement.

## I. INTRODUCTION

Flying Ad Hoc Networks (FANETs) are a variant of Wireless Ad Hoc Networks, consisting of small Unmanned Aerial Vehicles (UAVs) [1], [2]. These networks do exhibit widespread usage in the field of military, commercial and civilian applications. Indeed, using multiple UAVs extend the field of application to scenarios characterized by longer mission durations and wider areas to be covered. A number of novel and unique features associated to the support of mobility, distributed control without need for central coordination, and the increased communication range can be supported using drones' swarms. This makes these networks of paramount importance when ordinary communication infrastructure is not available, for example in case of catastrophic situations [3]–[5] or in remote areas not connected with the structured Internet [6], [7]. Indeed a rapidly deployable and highly flexible infrastructure can be set up by means of designing a robust communication architecture and protocol set to follow highly dynamic network variations. The use of multi-UAV systems

offers also a great advantage in terms of scalability, as it allows us to cope with the need for providing resources in networks of increasing size [8]. To deal with this call for scalability, in a multi-UAV system some UAVs can also rely on a ground base and/or satellite so leading to a variable topology, not only a star one, but a meshed one. However, the realization of a reliable multi-UAV system implies solving a number of design issues which transversely involve both hardware and software aspects. On the one hand, indeed, devices should be equipped with expensive and complex hardware to communicate among themselves and with a ground base or satellite. On the other hand, concerning software features, critical aspects are related to the capability to support dynamic and time-variant topological and environmental conditions associated to the high speed of devices, node movements and terrain structures. In addition, it should be considered that UAVs might not keep their communications links always on because they are battery-powered, and network lifetime should be increased as much as possible.

At the same time, the new generation of communication networks (5G), by employing various changes in network architecture and new technologies, is emerging to enable key performance indicators (KPI) in terms of high capacity, enhanced data rate, very low latency, and flexible and scalable networking to allow users to receive an acceptable level of Quality of Experience (QoE) for new vertical applications [9], [10].

FANETs are considered as a valid tool to empower the 5G network architecture [11]–[13] because, in spite of the structural limited capabilities of the devices (e.g. limited available energy to supply engines and computing elements and limited computing capabilities), employment of multiple small UAVs can allow accomplishing complex mission tasks. In particular, in the perspective of a 5G infrastructure, UAVs can provide ubiquitous connectivity and computation for different device types also located remotely, for example in rural areas, in the framework of agricultural applications [7]. Specifically, integration of UAVs into a macro-cell network is drawing novel interest in the view of supplementing terrestrial cellular networks. In integrating FANETs into the 5G architecture, the limited capabilities of the hardware/software UAV resources should be exploited in the best way, while coping simultaneously with management and coordination aspects and

supporting services to 5G users.

In this paper, we present a comprehensive resource management framework for placement of service chains. It is based on a mathematical optimization model that allows us to represent a complex dynamic and totally distributed scenario, where FANETs are integrated into the 5G ecosystem with the aim of providing services in terms of virtual network and application functions, in the sequel referred to as *Virtual Functions* (VF), running inside the FANET UAVs and organized as service chains. In doing this, we take into account not only the service requirements, but also the hardware device limitations in terms of scarceness of energy to supply UAV engines and computing elements. Regarding this second term, we highlight the consumption due to keep active the computing element, if needed, the consumption to keep alive VFs running to support even a single flow, and the additional amount of energy consumed to support other flows that can share the same function. Thus, the main contributions in this paper are the following:

- Present a system architecture based on the use of FANETs to provide remote areas with a 5G softwarized network where running service chains of VFs with strict requirements in terms of maximum tolerated delays.
- Propose a mathematical optimization model, allowing VF placement and chaining, aimed at minimizing the energy consumption and the no-service and partial-service probabilities of the FANET as a whole, without employing heuristics for the solution of the problem. The proposed model takes into account the processing capabilities available on board each UAV, and the transmission rate of the communications links between UAVs.
- Introduce two VF placement strategies, named *MemoryLess Placement* (MLP) and *With-Memory Placement* (WMP), the former aimed at saving more energy, the latter at minimizing flow re-routing after each FANET modification event.
- Propose an extensive numerical analysis to compare MLP and WMP with a standard strategy, referred to as *No-Share Placement* (NoShP), which does not share VFs among different flows and does not take into account the packet-rate and bit-rate input/output modifications caused by each VF.

The rest of this paper is organized as follows. In Section II we discuss some related work in the field. In Section III we describe the system and in Section IV we model it. In Section V the problem is formulated. Then, in Section VI we present a numerical analysis also comparing the performance of our solution with a standard approach. Finally, in Section VII conclusions and some considerations on the future work are presented.

## II. RELATED WORK

Unmanned aerial networks as discussed by Khan et al. [14] and Zeng et al. [15], are today envisaged to be used in numerous application scenarios, ranging from disaster management

in hostile environments, to search and rescue operations, border surveillance, wildfire management and civil security [16]–[24]. Other application scenarios include agricultural remote sensing [25] and traffic monitoring [26], as also discussed by Khan et al., Zeng et al., and Liu et al. in [14], [15], [27].

Depending on the specific application scenario, various communication architectures can be considered. Indeed, due to the rapid variations in network topology and the corresponding associated routing protocol issues coming from this dynamicity, network updates should be sophisticated enough to cope with this flexibility. Also the use of a peer-to-peer network topology is needed to support collaborative coordination among peers, so as to provide services and functions to be delivered to on-demand users and devices on the ground. This way, FANETs can be seen as providers of services and/or slices for 5G networks. In [14], authors Khan et al. illustrate different UAV architectures, each one characterized by different peculiarities in terms of tradeoff between simplicity, vulnerability, maximum network size to be supported, and robustness.

When a FANET is employed for collaborative purposes, the design objective is to support cooperative communications among multi-UAVs. Due to the devices' mobility in time, collaborative scheduling calls also for an efficient setup of routing protocols (e.g. static, proactive, reactive, hybrid, geographic/position-based or hierarchical routing protocols), as addressed for example in [14], [15]. Similarly, in [28], due to network variations and local topology changes during UAVs' aerial fleet flight, the focus of Han et al. is put specifically on solving the problem of multipath routing transmission in terms of multipath establishment and maintenance. Also in [29], Rosati et al. deal with the issue of maintaining communication links between the UAVs by also providing a comparison between two different routing algorithms for ad-hoc networks, such as the optimized link-state routing (OLSR) and the predictive OLSR (P-OLSR), which exploit GPS information available at drones. Authors Zheng et al. in [30] move in the direction of considering simultaneously routing and medium access control issues to improve the communication performance of FANETs.

The problem of employing UAVs for expanding and complementing existing cellular infrastructures according to a 5G perspective was considered in [31]–[37]. However, in [34] Mozaffari et al. use UAVs as flying base stations to provide a wireless backbone and give coverage to a remote area. The perspective is only mathematical, in the view of providing a tractable analytical framework for the coverage and rate analysis. Also in [31]–[33], the main focus was on optimizing connectivity, and solving coexistence issues in case of underlying Device-to-Device (D2D) communication links, while also compensating cell overload and outage in cellular networks. A slightly different perspective is considered by Mozaffari et al. in [38] where the UAVs are considered as aerial base stations to collect data from the terrestrial Internet of Things (IoT) or to provide connectivity to IoT devices, for example in power plants, as proposed by Faraci et al. in

[39]. In [38] the authors Mozaffari et al. specifically present a framework for joint optimization of UAVs positioning in a 3D scenario and to perform controlled device-UAV association, as well as power control in uplink to minimize the total power consumption. In [42] the authors Mohamed et al. propose the idea of implementing a UAVFog for rapid replacement of connection between the IoT network and the previously available fog platforms, while taking into account provision of different low-latency services to support IoT applications at certain areas.

In [40] authors present a framework for managing and orchestrating UAVs' services in MEC NFV environments. The framework is aimed at joining MEC-NFV management and UAV Traffic Management. The proposed system is modeled using an Integer Linear Programming (ILP) approach to minimize deployment costs while satisfying QoS requirements. Similarly, in [41] the problem of supporting low latency service deployment over a network with dynamic topologies complemented by means of mobile nodes (e.g., robots, drones) is discussed. The VNF placement problem is presented and dealt with as a cost-minimizing problem. In addition, a heuristic is proposed to solve the problem. Also a heuristic is proposed to solve the problem. However, only recently few papers have appeared that address the problem of processing complex computational tasks with the help of drones to assist specific missions. In this case some complex tasks, for example, cannot be offloaded to the edge and cloud infrastructure. This could happen because some missions can be executed in remote areas where there is lack of reliable access to the edge or cloud resources, while requiring processing of some tasks with urgency.

In this view, it has been recently proposed to exploit light-weight drone facilities, by providing virtualized computing capabilities made available by single-board computers as proposed by Alharthi et al. in [43] and Faraci et al. in [6]. Similarly, this was also discussed by Rametta et al. in [44] for supporting video monitoring applications in wide rural areas, not covered by Internet access.

According to a NFV perspective, a flexible and dynamic connectivity backbone can be provided by a set of drones. The main issue is, thus, associated to the realization of appropriate service chains which allow execution of complex network and application functions, by exploiting a combination of simpler virtual functions; in this way the Orchestrator has to decide the backbone UAVs where running the virtual functions, and assign the corresponding resources to be allocated. The problem of virtual function allocation and service chaining has been addressed significantly in the context of traditional wired networks. In [45], Yi et al. discuss NFV with a focus on avoiding network ossification and difficulty in network management and service provisioning. Accordingly, by means of NFV, it is possible to decouple network functions from the dedicated hardware and providing improved flexibility in service provisioning. In [45] and references therein, the problem of optimal network function allocation is addressed by way of mathematical frameworks based on ILP or Mixed ILP

(MILP) [46], [47] with the aim of minimizing the Operating Expense (OPEX) and maximizing network utilization. Also in [47], an ILP problem is formulated and solved by Riggio et al. for optimal VNF placement solution under radio resource constraints. In [48], [49] similar problems are addressed by Luizelli et al. and Gupta et al., but as separate problems with a specific focus either on bandwidth minimization or reduction in end-to-end delay and resource waste.

However, relevant scalability issues arise in these solutions and too long convergence time is experienced in [50] by Pham et al., especially upon scaling to large network size which makes these mechanisms potentially unsuitable to real settings. One of the main common problems is that, in order to solve the mathematical framework, usually heuristic solutions are resorted [46], [47] because of their relevant convenience in terms of execution times, as well as for the good degree of approximation to mathematical solutions. In [51] the two opposite targets of minimizing the maximum network link utilization while considering a reduced number of CPUs for VNF instantiation was addressed by Addis et al.. The two problems were considered separately and then prioritized. Other works focused on the joint solution of the problem in the context of both physical and Virtual Network Functions (VNFs), like in [52] by Moens et al.. Other approaches, e.g. [53] by Khebbache et al., used multi-stage graphs but had limitations in the practical solution of the problem to only few VNFs per service chain.

The service-chaining problem has been considered as an NP-hard evolution of the problem of VNF placement. This implies not only deciding where to place appropriate functions into network nodes, but also suitably taking into account physical features of nodes storing these functions and steering traffic through these placed functions.

There are a lot of deterministic and heuristic methods proposed to solve the service-chaining problem. However, due to the increasing size of networks, heuristic methods are so far the most accepted approaches, because they make solution of the service-chaining problem reasonable in scenarios where deterministic solutions are unfeasible [46]. For example, in [50] a heuristic service chaining approach which can be solved within polynomial time is presented by Pham et al.. Some of these heuristics however focus on solution of partial problems, such as energy consumption reduction or reliability increase only, e.g. as proposed by Fan et al. in [54].

In [52] a hybrid problem where part of the services may be provided by dedicated physical hardware and another part by exploiting virtualized service instances is presented. This exhibits great advantages in terms of service deployment costs and resource utilization, although tested in small networks. In [55] Eramo et al. and in [56] Scheid et al. proposed other approaches based on network function migration and automated service chaining construction.

However, none of the above solutions deals with the joint problem of providing an exact and solvable mathematical framework for network function provisioning and service chaining, while taking into account both the physical features

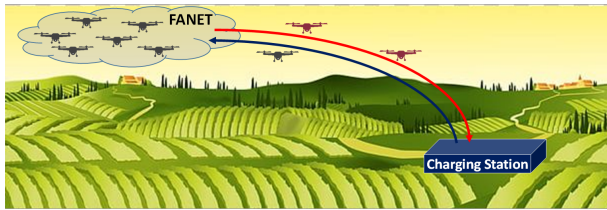


Fig. 1: A FANET providing a 5G softwarized network to a remote geographic area

of the UAVs like residual energy, weight, CPU and memory, storage features, or the possibility to have constraints in the framework in terms of delay on chains, as well as requirements in terms of energy consumption reduction, load balancing, packet loss probability minimization and failure probability reduction in service-chains placement.

The framework proposed in this paper is able to provide both a mathematical, completely-deterministic and flexible solution, without using any heuristic. It can be implemented in real scenarios where UAVs need to leave the network for recharging purposes but service provisioning should be still guaranteed. In Table I we report a summary of what has been discussed in this section.

### III. SYSTEM DESCRIPTION

Let us consider a fleet of UAVs organized as a FANET (see Fig. 1). This fleet provides a 5G softwarized network on demand to users and devices on the ground in a remote geographic area (see [57]). In the following, for the sake of simplicity, we will assume that all the UAVs are in each other coverage range, that is, each UAV is reachable from any other UAV of the FANET through a single-hop communication. However, extension to the more general case of multi-hop communication between UAVs in the same FANET is straightforward.

The wireless channel between each 2-uple of UAVs  $s$  and  $\bar{s}$  is characterized by its transmission rate,  $\bar{T}_{s\bar{s}}$ , that depends on many external factors, as the encoding technique, the bandwidth allocated to the channel, the distance between the UAVs, the transmit power, the average power of the channel noise, and the mutual interference, according to a specific channel model whose definition is out of the scope of this paper. An interested reader can find details regarding models of UAV wireless channels in [58], [59].

Each UAV of the FANET, as depicted in Fig. 2, is equipped with a Computing Element (CE) that is able to work as a NFV Infrastructure Point-of-Presence (NFVI-PoP), where not only network functions, but also application functions, can be run as Virtualized Functions (VF). VFs are used as building blocks of Service Chains (SC) to realize Network Services (NS) and Application Services (AS). For this reason, each CE has to be deployed with a resource virtualization environment (e.g. a hypervisor to run virtual machines or a docker/container engine to run Linux containers or dockers), and an SDN switch

(e.g. an open virtual switch) to chain local VFs with each other and with VFs running on other UAVs.

Placement of VFs and their chaining must be done taking into account the service requirements the network has to guarantee to its users, and some constraints imposed by the FANET infrastructure, in terms of CE computing power and transmission rate of wireless links among UAVs.

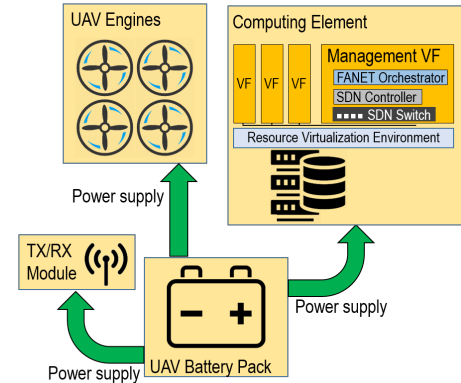


Fig. 2: Functional architecture of a UAV

As shown in Fig. 2, the battery installed on board the UAV is used to supply the UAV engines, the CE and the Transmitter/Receiver (TX/RX) module. Given the computing features of the CE, we cannot neglect the amount of energy it consumes, which is comparable with the one consumed by the engines [60] [61] [7]. Therefore, the presence of a switched-on CE installed on board the UAV and the execution of some VFs have a cost in terms of battery power consumption because this causes a reduction in the flight duration. Specifically, when the battery charge of a UAV is below a given threshold, the UAV has to temporarily leave the FANET to go to the nearest Charging Station. During this period, this UAV is unavailable, and the FANET has to be reconfigured placing the VFs that were running in that UAV in the remaining UAVs. As a consequence of it, we have an increase in the power consumption of the remaining active UAVs, and therefore a reduction in their flight duration, and a decrease of performance provided to the user flows. In some cases, it could happen that, if the number of flying UAVs is not sufficient, the FANET is not even able to provide users with all the required service chains.

For this reason, maximizing the UAV flight duration is a crucial feature in service provisioning. This is done minimizing both energy consumption of UAVs when they are working in the FANET and their outage service period for recharging.

In order to minimize the duration of this last period, we consider that enough backup batteries are available at the Charging Station, and a sufficient number of electrical plugs are installed there. This way, upon landing, the UAVs always find an available battery to substitute the discharged one to takeoff immediately in order to reach again the FANET. So, their outage service period consists only in the time to reach the Charging Station, substitute the battery (assuming that

TABLE I: Related Work

Ref.	Application scenario	Contributions and Challenges	Type of contribution
[6]	Integration of IoT and fog in remote areas	Minimize power consumption, decide the number of active CPUs at runtime	Reinforcement learning, dynamic programming
[24]	Disaster management and civil security applications	Distributed task allocation, conflict resolution, plan refinement	Experimental work, no mathematical modeling
[25]	Agricultural remote sensing	Mission planning, flight command activation and flight plan monitoring, develop a navigation system, temporal monitoring of the system	Experimental work, no mathematical modeling
[26]	Traffic monitoring in complex urban environments	Control occlusions for UAV surveillance, trajectory design and control	Algorithmic approach with simulations in AgentFly software
[28]	5G ecosystems	Multipath routing scheme design based on network coding and ant colony optimization, reduce the routing reconstruction overhead	Mathematical framework, pheromone computation algorithm
[29]	5G ecosystems	Field-tests used to compare routing protocols (P-OLSR vs. OLSR), including MAC-layer emulations	Experimental work, no mathematical modeling
[30]	5G ecosystems	Hybrid communication protocol including position-prediction-based directional MAC protocol (PPMAC) and a self-learning routing protocol based on reinforcement learning (RLSRP)	Matlab and NS2 simulations
[31]	Homeland security including control of incident areas	Efficient agent algorithms for a UAV-swarm for the transmission of both payload (sensor) information and real-time critical telemetry	Experimental work, no mathematical modeling
[32]	5G ecosystems	Optimize the location and movement of UAVs to improve the connectivity. Four types of network connectivity are considered (global message connectivity, worst-case connectivity, network bisection connectivity, and k-connectivity)	Mathematical framework, Heuristics
[33]	Offloading cellular network traffic	UAV-driven aerial cell overload and outage compensation in LTE networks, frequency planning, relay node placement	Analytical model
[37]	5G ecosystems	Dynamic resource activation, VNF placement, CPU	Mixed-Integer Non-linear Programming (MINLP), Heuristics
[38]	Aerial BS to collect data from IoT	Optimization of UAVs positioning, Power control	Analytical model
[39]	Power plants, smart industry	Support best connectivity over time considering the variability of both the bandwidth request and the wind energy	Reinforcement learning
[40]	5G ecosystems	Management and orchestration of UAV services in MEC NFV networks, support of QoS requirements	ILP
[41]	5G ecosystems and mobile fogs	Low latency service deployment, cost minimization problem	Heuristics
[42]	5G ecosystems and mobile fogs	UAV for replacement of connection between IoT devices, support of low latency services	Experimental work, no mathematical modeling
[43]	5G ecosystems	Providing virtualized computing capabilities in single-board computers	Mathematical framework
[44]	Video monitoring in rural areas	Dynamic selection of video transmitters decided by users according to the portions of rural areas they want to monitor, definition of the service chains to realize the video delivery service, evaluate the computational load of the platform nodes	Analytical model
[46]	5G ecosystems	Showing that dynamic VNF orchestration can have more than 4x reduction in OPEX compared to hardware middleboxes	ILP and Mixed ILP, Heuristics
[47]	5G ecosystems	VNF placement problem for radio access networks, a slice scheduling mechanism that ensures resource and performance isolation between different slices	ILP and Mixed ILP, Heuristics
[48]	5G ecosystems	Bandwidth minimization, formalization of the network function placement and chaining problem	Heuristics
[49]	5G ecosystems	Reduction in end-to-end delay and resource waste	ILP
[50]	5G ecosystems	Joint operational and network traffic cost as an optimization problem whose goal is to minimize the total cost of the network	Markov approximation (MA), combinatorial problem and heuristics
[51]	5G ecosystems	Minimize the maximum network link utilization, VNF Placement and Routing optimization problem	Heuristics and meta heuristics
[52]	5G ecosystems	Both physical and Virtual Network Functions placement problem	ILP
[53]	5G ecosystems	Multi-stage graphs for optimization, chained service graph for VNF placement	Multi-stage graph representation, heuristics
[54]	5G ecosystems	Energy consumption reduction, reliability minimization	Analytical framework
[55]	5G ecosystems	Network function migration	ILP
[56]	5G ecosystems and enterprise networks	Automated service chaining construction, service chaining management according to Service Level Agreements (SLAs)	Graphs and controlled natural language

there is an automatic system requiring no human intervention [62]), and the time to join again the FANET.

On the other side, minimizing energy consumption during service provision is a crucial issue to maximize FANET performance. Optimization of this task is the main goal of this paper. It is entrusted to a central entity called *FANET Orchestrator* that runs in one of the UAVs belonging to the FANET, playing the role of *FANET Coordinator*. At the FANET set up or after each critical event (e.g. when the FANET Coordinator leaves the FANET to go to the Charging Station), this role is transferred to another UAV. Policy definition for assignment of this role is out of the scope of this paper.

The FANET Orchestrator function is run in the *Management VF* together with the local SDN Switches and an SDN Controller, the latter with the role of controlling the SDN Switch of all the FANET UAVs. Note that the Management VF is always running in each flying UAV to maintain the SDN Switch active. In addition, in the FANET Coordinator, the same VF assumes the roles of FANET Orchestrator and SDN Controller. This VF is switched off only when the UAV leaves the FANET to reach the Charging Station.

We can assume that execution of the FANET Orchestrator function does not influence power consumption of the Management VF because this function is run on a larger time scale (dozen of minutes) than the time scale of network packet management. Moreover, consider that, although each optimization run does not require too much time to be executed, it is possible to run it offline on an external data center and save results in a table to be retrieved instantaneously when needed by UAVs. The same holds for the SDN Controller, whose action is required only during FANET placement reconfiguration. For this reason, in the estimation of the energy consumption of the Management VF, we will neglect the contribution of the two above functions, and only consider the local SDN Switch, which is present in all the active UAVs in the FANET.

The VF placement and chaining strategy will be applied each time the FANET conditions change. This occurs for one of the following reasons:

- 1) the FANET topology is changed because of unavailability of one or more UAVs, or in case a new UAV enters the FANET after battery replacement;
- 2) the input traffic has considerably changed;
- 3) new service chains are requested to the FANET, either using the same VFs, or requiring the activation of some new VFs.

Therefore, in the following, we will focus on the procedure run at the beginning of a time period immediately after a change in FANET conditions. In this period, the FANET is required to provide a given number of services (either NSs or ASs), each implemented as a chain of VFs.

Let each SC be composed of an arbitrary ordered sequence of VFs. Fig. 3 shows an example of SCs; the same example will be considered in the use case presented in Section VI. We will represent the aggregation of flows requiring the same SC as a single flow. It is the superposition of many flows generated by different devices and having different destinations, but

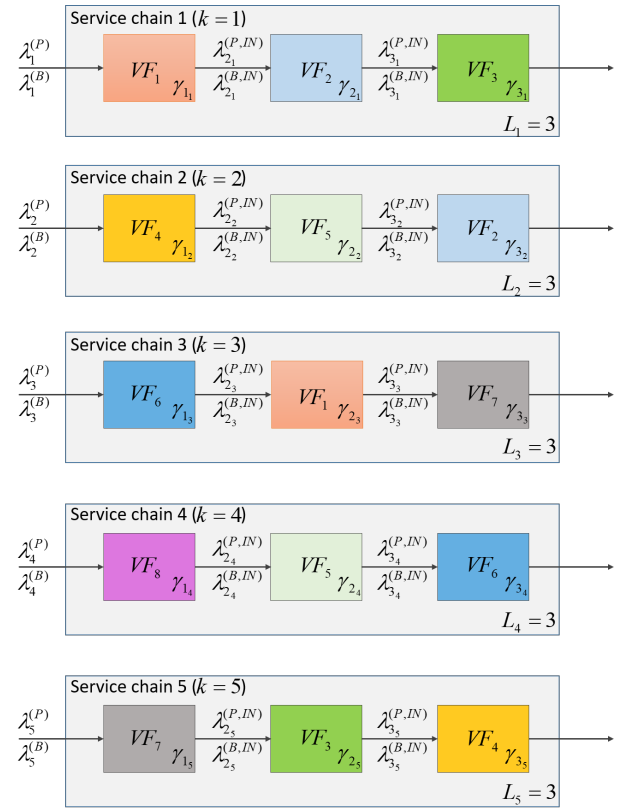


Fig. 3: Service chains considered in our scenario.

all requiring the same SC. As shown in Fig. 3, each VF can belong to more than one SC (e.g. the  $VF_1$  belongs to the first and the third service chains). This way, each VF instance is used by one or shared by more flows. Each flow is characterized by its mean bit rate and its mean packet rate, respectively indicated as  $\lambda_k^{(B)}$  and  $\lambda_k^{(P)}$ , for  $k \in [1, 5]$  in Fig. 3. As compared to other papers proposed in the literature, in this paper we consider the more general case where, as shown in Fig. 3, VFs can modify flow rates because of their specific peculiarities; accordingly, we assume that the output packet rate and/or the output bit rate can be different from the input ones. For example, a Firewall, which can discard packets according to the implemented rules, modifies the packet rate by decreasing it, or an Encrypter can maintain the packet rate constant, but increases the bit rate by introducing some redundancy.

The CE of each UAV is shared by all the VF instances that run on it. We will assume that sharing is obtained by fairly dividing the whole computing power of the CPU of a CE in a number of slices equal to the number of running VF instances. No priority among slices is considered in this paper, but extension to introduce some prioritization is trivial. To achieve isolation among different instances running on the same CE, as shown in Fig. 4, one first-in-first-out (FIFO) queue is associated with each VF instance to enqueue packets that cannot be served immediately when they arrive to the CE. Specifically, Fig. 4 represents the case of three VFs running

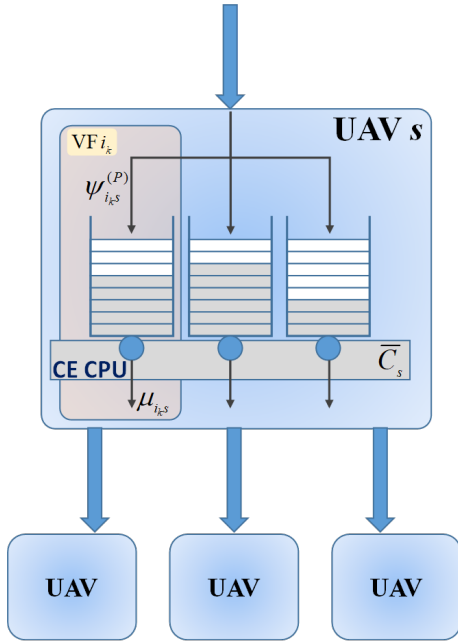


Fig. 4: UAV internal structure in case of three VFs sharing the same CE CPU

in the same UAV, sharing the CPU capacity  $\bar{C}_s$ .

In the following sections, we will model the FANET described so far in order to decide the optimum placement that minimizes an objective function weighing the overall power consumption of the FANET and a cost contribution capturing situations when the FANET is not able to provide the required services. Besides, in order to consider a set of constraints accounting for the physical limitations of the FANET components (e.g. the CPU computation power of the CE installed on board each UAV and the maximum transmission capacity of the wireless links connecting UAVs), we will add a Quality of Service (QoS) constraint regarding the maximum end-to-end delay for each service chain.

Two placement strategies will be considered in our framework: MemoryLess Placement (MLP) and With-Memory Placement (WMP). They are run by the FANET Coordinator at each FANET modification event, i.e. when a UAV leaves the FANET to change its battery, or when a UAV comes back to the FANET after battery replacement.

The *MLP strategy* is memoryless because, when placement optimization is run, it does not consider the previous position of VFs on UAVs before the FANET modification event.

The *WMP strategy*, on the other hand, when placement optimization is run, starts from the placement before the FANET modification event, and decides the new placement optimizing the same objective function used by MLP, but with some additional constraints imposed to maintain as much as possible the previous placement of VFs. More specifically, when a UAV leaves the FANET, optimization is applied to decide placement of the only VFs that were running on that UAV. Instead, when a UAV comes back to the FANET, the

WMP strategy calculates the optimal placement of the VFs that were placed on that UAV only, when all the UAVs were available in the FANET.

The above two strategies will be described in details in Algorithms 1 and 2, in Section V-C4. Let us note that, since WMP adds some constraints to the problem formulated for MLP, it obtains a worse objective function, but with the advantage of reducing flow re-routing after each FANET modification event.

#### IV. SYSTEM MODEL

We model a FANET as a graph  $G = (V, E)$ , in which:

- $V$  is the set of UAVs available in the FANET; let  $s \in V$  be the generic UAV;
- $E$  is the set of point-to-point wireless transmission links connecting UAVs; let  $l_{s\bar{s}} \in E$  be the generic link between UAV  $s$  and UAV  $\bar{s}$ .

Also, we define:

- $K$  as the number of SCs to be provided by the FANET; let  $k$  be the generic SC;
- $F$  as the set of available VFs; let  $f \in F$  be the generic VF;
- $f_k$  as the VF  $f$  in the service chain  $k$ ;
- $L_k$  as the number of VFs that constitute the  $k$ -th service chain.

In order to capture the behavior of VFs that modify the packet rate and the bit rate of the input flows, we define:

- $\lambda_k^{(P)}$  and  $\lambda_k^{(B)}$ : the input packet rate and the input bit rate of the aggregated flow using the service chain  $k$ ; if we indicate the mean packet size as  $\Omega$ , we have:

$$\lambda_k^{(B)} = \Omega \cdot \lambda_k^{(P)}; \quad (1)$$

- $\gamma_{jk}^{(P)}$ : the mean value of the ratio between the output packet rate and the input packet rate of the  $j$ -th VF in the SC  $k$ . Note that  $\gamma_{jk}^{(P)}$  can assume any positive value. For example, a Firewall is characterized by  $\gamma_{jk}^{(P)} < 1$  because it reduces the packet rate, a Traffic Monitor has  $\gamma_{jk}^{(P)} = 1$  because it does neither add nor discard any packet, while a Multicast Packet Replicator is characterized by  $\gamma_{jk}^{(P)} > 1$  since it introduces new packets;
- $\gamma_{jk}^{(B)}$ : the mean value of the ratio between the output flow bit rate and the input flow bit rate of the  $j$ -th VF in the SC  $k$ .

Note that the values of  $\gamma_{jk}^{(B)}$  and  $\gamma_{jk}^{(P)}$  can be different from each other. For example, for a VF that adds some redundancy to each packet, the bit rate is increased whilst the packet rate

remains unchanged<sup>1</sup>.

The generic flow that enters the SC  $k$  with an input bit rate  $\lambda_k^{(B)}$  and an input packet rate  $\lambda_k^{(P)}$ , at the input of the  $i$ -th VF of the same chain has a bit rate given by:

$$\lambda_{i_k}^{(B,IN)} = \lambda_k^{(B)} \prod_{j_k=1}^{i_k-1} \gamma_{j_k}^{(B)} \quad (2)$$

and a packet rate given by:

$$\lambda_{i_k}^{(P,IN)} = \lambda_k^{(P)} \prod_{j_k=1}^{i_k-1} \gamma_{j_k}^{(P)}. \quad (3)$$

Our goal is to formulate a problem to decide the VF placement minimizing the total power consumption. In particular, we consider the total power consumption as composed of the following elements:

- $p_s^{(E)}$ , the power consumption of the UAV  $s$  necessary to supply its engines; this represents the cost of having a UAV available in the FANET, but inactive, i.e. not participating in service provisioning;
- $p_s^{(CE)}$ , the power consumption to keep a CE active in the UAV  $s$  (without any running VM, and therefore any VF);
- $p_{f_s}^{(VM)}$ , the power that the VM hosting the instance of the VF  $f$  deployed and running on the UAV  $s$  consumes to be active, not considering flow processing;
- $P_{i_k s}^{(PR)}$ , the power consumed by the  $i$ -th VF of the SC  $k$  for packet processing; it depends on the energy needed by the same VF to process one packet,  $e_{i_k s}$  as follows:

$$P_{i_k s}^{(PR)} = e_{i_k s}^{(P)} \cdot \lambda_{i_k}^{(P,IN)}, \quad (4)$$

where  $\lambda_{i_k}^{(P,IN)}$  is the input packet rate of the considered VF;

- $P_{s\bar{s}}^{(TX)}$ , the power consumed by the UAV  $s$  to transmit to the UAV  $\bar{s}$ . In case that  $s = \bar{s}$ , it follows that  $P_{s\bar{s}}^{(TX)} = 0$ .

Each queue associated with a VF instance (see Fig. 4) is modeled as a M/M/1 system, which is a single-server queueing system with Poisson-distributed arrivals and exponentially-distributed packet service times. The mean arrival packet rate to the  $i$ -th VF of the SC  $k$  running on the UAV  $s$ , here indicated as  $\psi_{i_k s}^{(P)}$ , is the sum of the packet rate of all the flows using that VF. Since that VF can belong to other SCs besides the  $k$ -th one, if we indicate the generic SC using that VF as  $h$  and the position of that VF on the SC  $h$  as  $G_{i_k h}$ , the aggregated packet rate entering that VF is:

$$\psi_{i_k s}^{(P)} = \sum_{h=1}^K x_{(G_{i_k h})s} \lambda_{G_{i_k h}}^{(P,IN)} \quad \forall k, i_k, s, \quad (5)$$

<sup>1</sup>Although in the definition of  $\gamma_{j_k}^{(P)}$  and  $\gamma_{j_k}^{(B)}$  we have explicitly expressed the dependence on the position  $j$  in the chain  $k$  of the referred VF, actually they depend on the kind of VF, whatever its position in the SC. For this reason, in the use case illustrated in Section VI, they will be provided as a function of the VF  $f$ ,  $\forall f \in F$ , rather than of its position in a specific chain. The same will be done for the variables  $P_{i_k s}^{(PR)}$ , representing the power consumed by the  $i$ -th VF of the SC  $k$  for packet processing, and  $e_{i_k s}^{(P)}$ , representing the same VF to process one packet. In Section VI, they will be indicated as  $P_{f_s}^{(PR)}$  and  $e_{f_s}$ .

where  $x_{(G_{i_k h})s}$  is a Boolean variable indicating whether the VF that is in position  $G_{i_k h}$  in the chain  $h$  is deployed in the UAV  $s$  or not. It will be formally defined in Section V-A, specifically in (8).

Let  $\mu_{i_k s}$  be the service rate of that queue, that is, the number of packets that the  $i$ -th VF of the chain  $k$  is able to process in the time unit in the UAV  $s$ . If we indicate the number of elementary operations required to process one packet by this VF instance as  $\sigma_{i_k s}$ , the number of VF instances running in the same UAV  $s$  as  $N_s^{(VF)}$ , and the whole CPU computation capacity of the same UAV (i.e. the number of elementary operations that the CE CPU is able to compute in the time unit) as  $\bar{C}_s$ , the service rate  $\mu_{i_k s}$  can be calculated as follows:

$$\mu_{i_k s} = \frac{\bar{C}_s}{N_s^{(VF)} \sigma_{i_k s}}. \quad (6)$$

According to the M/M/1 queueing theory, the average response time or sojourn time (i.e. total time a packet spends in the M/M/1 queueing system) is:

$$r_{i_k s} = \frac{1}{\mu_{i_k s} - \psi_{i_k s}^{(P)}}. \quad (7)$$

A summary of used notations is presented in Table II.

## V. PROBLEM FORMULATION

In this section, we provide a mathematical formulation of the considered system and propose a VF placement and chaining model aimed at minimizing the power consumption of the FANET as a whole. The main notation used in this section is listed in Table III.

### A. Variables

Let  $x_{i_k s}$  and  $w_{f_k s}$  be two Boolean variables defined as:

$$x_{i_k s} = \begin{cases} 1 & \text{if the } i\text{-th VF of the service chain } k \text{ is} \\ & \text{assigned to the UAV } s \end{cases} \quad (8)$$

$$w_{f_k s} = \begin{cases} 1 & \text{if the VF } f \text{ of the service chain } k \text{ is} \\ & \text{assigned to the UAV } s \end{cases} \quad (9)$$

It is evident that, if the VF  $\tilde{f}$  in the chain  $\tilde{k}$  is in position  $\tilde{i}$ , the following equality applies:  $x_{\tilde{i} \tilde{k} s} = w_{\tilde{f} \tilde{k} s}$ ,  $\forall s$ .

Moreover, let us define:

- the number of service chains that share the same VF  $f$  in the same UAV  $s$ :

$$N_{f_s}^{(SC)} = \sum_{k=1}^K w_{f_k s}; \quad (10)$$

- the number of VF instances running on the UAV  $s$ :

$$N_s^{(VF)} = \sum_{k=1}^K \sum_{f \in F} w_{f_k s}. \quad (11)$$



TABLE II: Basic notations regarding the system model

Symbol	Definition
$s \in V$	The generic UAV
$l \in E$	The generic point-to-point link between two UAVs
$k \in K$	The generic SC
$f \in F$	The generic VF
$f_k$	The VF $f$ in the Service Chain $k$
$L_k$	Number of VFs in the $k$ -th chain
$\lambda_k^{(P)}$	Input packet rate of the $k$ -th chain
$\lambda_k^{(B)}$	Input bit rate of the $k$ -th chain
$\Omega$	Mean packet size
$\gamma_{i_k}^{(P)}$	Ratio between the output packet rate and the input packet rate of the $i$ -th VF in the chain $k$
$\gamma_{i_k}^{(B)}$	Ratio between the output bit rate and the input bit rate of the $i$ -th VF in the chain $k$
$\lambda_{i_k}^{(B,IN)}$	Bit rate at the ingress of the $i$ -th function in the $k$ -th chain
$\lambda_{i_k}^{(P,IN)}$	Packet rate at the ingress of the $i$ -th function in the $k$ -th chain
$p_s^{(E)}$	Power consumed by the engines of the UAV $s$
$p_s^{(CE)}$	Power consumed to keep active the CE of the UAV $s$
$p_{f_s}^{(VM)}$	Power consumed to maintain the VM $f$ in the UAV $s$ switched on
$e_{i_k s}^{(P)}$	Energy consumed by the $i$ -th VF in the chain $k$ to process one packet
$e_{s\bar{s}}^{(B)}$	Energy consumed for transmission of one bit on link between two generic UAVs $s$ and $\bar{s}$
$P_{i_k s}^{(PR)}$	Power consumed by the $i$ -th VF of the SC $k$ for packet processing
$P_{s\bar{s}}^{(TX)}$	Power consumed by the UAV $s$ to transmit to the UAV $\bar{s}$
$G_{i_k h}$	Position of the $i$ -th VF on the SC $h$
$\psi_{i_k s}^{(P)}$	Aggregated packet rate entering the VF in position $G_{i_k h}$
$\sigma_{i_k s}$	Number of elementary operations required by the VF $f$ in the UAV $s$ to process one packet
$\bar{C}_s$	CPU computation capacity of the CE in the UAV $s$
$\mu_{i_k s}$	Service rate of the processing queue of the VF $i_k$ in the UAV $s$
$r_{i_k s}$	Total time spent by a packet in the queue of the VF $i_k$ in the UAV $s$
$\bar{T}_{s\bar{s}}$	Maximum link transmission rate between two generic UAVs $s$ and $\bar{s}$
$\bar{D}_k$	Maximum end-to-end delay for the chain $k$
$d_{s\bar{s}}$	Propagation delay on the link between two generic UAVs $s$ and $\bar{s}$

In order to represent if the UAV  $s$  is used to provide at least one SC, we define the variable  $z_s$  as follows:

$$z_s = \begin{cases} 1 & \text{if } N_s^{(VF)} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Similarly, we introduce  $z_{f_s}$  to represent whether the UAV  $s$  is used to run the VF  $f$  for at least one SC or not:

$$z_{f_s} = \begin{cases} 1 & \text{if } N_{f_s}^{(SC)} \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

Moreover, we define the variable  $y_k$  to represent, in the objective function that will be described in (22), which VFs are more convenient to place in case some SCs cannot be accommodated due to insufficient resources (i.e. when too

TABLE III: Notation used in the problem formulation

Symbol	Definition
$x_{i_k s}$	Boolean variable indicating if the $i$ -th VF of SC $k$ is assigned to UAV $s$
$w_{f_k s}$	Boolean variable indicating if the VF $f$ of SC $k$ is assigned to UAV $s$
$N_{f_s}^{(SC)}$	Number of SCs that share the VF $f$ in the UAV $s$
$N_s^{(VF)}$	Number of VF instances running on the UAV $s$
$z_s$	Boolean variable indicating if UAV $s$ is used to provide at least one SC
$z_{f_s}$	Boolean variable indicating if UAV $s$ is used to run the VF $f$ for at least one SC
$y_k$	Boolean variable indicating if the chain $k$ is deployed
$y_{f_s}$	Describe if the function $f$ in the UAV $s$ is shared by at least two SCs
$R_{mt}$	Ratio between the memory used by CPLEX and the computational time to solve the problem
$R_{ts}$	Ratio between the computational time to solve the problem and the number of solutions provided by CPLEX
$R_{ms}$	Ratio between the memory used by CPLEX and the number of solutions provided by it
$P_E$	The total power consumed by the FANET for the engines
$P_{CE}$	The total power consumed by the FANET for maintaining the CEs in active state
$P_{VM}$	The total power consumed by the FANET for maintaining the VM instances switched on
$P_{VF}$	The total power consumed by the FANET to manage traffic
$P_L$	The total power consumed by the FANET for transmission on links
$G_{SF}$	Gain achieved by serving the number of chains that maximize the flow of services provided

many UAVs have temporarily left the FANET for battery replacement). More specifically, the variable  $y_k$  is a Boolean variable indicating whether the chain  $k$  has been placed in the FANET or not, that is:

$$y_k = \begin{cases} 1 & \text{if the chain } k \text{ has been placed} \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Finally, it is necessary to introduce an additional variable useful for model resolution: in order to describe the case in which the VF  $f$  is shared by at least two SCs on the UAV  $s$ , we define the Boolean variable  $y_{f_s}$  as follows:

$$y_{f_s} = \begin{cases} 1 & \text{if } N_{f_s}^{(SC)} \geq 2 \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

### B. Objective Function

The objective function allows us to minimize a penalty, due to the overall power consumption, and to maximize a gain, due to the overall packet rate that is served for the flows whose requirements are satisfied. Therefore, the objective function depends on two main terms: the penalty (or power consumption) and the gain (or users' satisfaction). More specifically, in order to maximize the aforementioned gain term, it will be considered with a negative sign in the objective function to minimize.

The first term of the objective function, that is the overall power consumption, is composed by the contribution of the UAVs' engines, the one for maintaining the CE, including the TX/RX Module and the Resource Virtualization Environment

in active state, the one to maintain the instances of VF containers (e.g. VMs, Docker or LXC) switched on, and the one to transmit data flows on the wireless links. We now present each of these components in detail.

The overall power consumption of the UAVs' engines is equal to the sum of the powers used by the engines of all the UAVs that are active in the FANET, namely:

$$P_E = \sum_{\forall s \in V} z_s \cdot p_s^{(E)}. \quad (16)$$

The overall power consumption for maintaining the CE, including the TX/RX Module and the Resource Virtualization Environment in active state is equal to the sum of the powers consumed to keep active the CE of each used UAV:

$$P_{CE} = \sum_{\forall s \in V} z_s \cdot p_s^{(CE)}. \quad (17)$$

The overall power consumption to maintain the VF containers switched on, not including their load by traffic flows (this will be considered in the next item), is given by the sum of the powers consumed to maintain the VF containers switched on for each running VF in each used UAV (for at least one SC):

$$P_{VM} = \sum_{\forall s \in V} \sum_{\forall f \in F} z_{fs} \cdot p_{fs}^{(VM)}. \quad (18)$$

The overall power consumption to manage traffic can be expressed as the sum of all the the powers  $P_{i_k s}^{(PR)}$  for the packet processing, as defined in (4), consumed by all the VFs of all the SCs that have been allocated to all the UAVs of the FANET. It can be derived as follows:

$$P_{VF} = \sum_{\forall s \in V} z_s \sum_{k=1}^K \sum_{i_k=1}^{L_k} P_{i_k s}^{(PR)}. \quad (19)$$

The overall power consumption for transmission on links between UAVs is calculated as the sum of the powers consumed by each UAV  $s$  to send bits to the following UAV  $\bar{s}$ :

$$P_L = \sum_{\forall s \in V} \sum_{k=1}^K \sum_{i_k=1}^{L_k-1} x_{i_k s} \sum_{\forall \bar{s} \in V} x_{(i+1)_{k\bar{s}}} \cdot e_{s\bar{s}}^{(B)} \cdot \lambda_{(i+1)_k}^{(B,IN)}. \quad (20)$$

Note that with the expression "the following UAV" we mean the UAV where the VF after the  $i$ -th one is placed. Moreover, if  $s = \bar{s}$ , it follows that  $e_{s\bar{s}}^{(B)} = 0$ .

The second term of the objective function is the gain achieved by serving "Satisfied Flows" (SF), i.e. flows whose SC is placed. This gain is proportional to the packet rate of SFs as follows:

$$G_{SF} = \sum_{k=1}^K \lambda_k^{(P)} y_k, \quad (21)$$

that is the sum of the input packet rates of the chains that have been placed.

The whole objective function, weighing the overall power consumption in the FANET and the gain for satisfied flows, is:

$$\Theta = \alpha_P \cdot (P_E + P_{CE} + P_{VM} + P_{VF} + P_L) + \quad (22) \\ - \alpha_G G_{SF},$$

where  $\alpha_P$  and  $\alpha_G$  are timely-chosen constants to provide more or less importance to the first or the other term.

Finally, the optimization problem is formulated as follows:

$$\min \Theta \quad (23)$$

s.t. constraints defined in (24)-(35).

### C. Constraints

In order to meet our objective, some constraints need to be satisfied. The first set of constraints, described in Section V-C1, are associated to the model variables. Two more constraints are described in Section V-C2, and regard limitations to the underlying FANET infrastructure. The constraint described in Section V-C3 specifies, at the application layer, an upper bound on the end-to-end delay for each chain. Finally, the last four constraints, defined in Section V-C4, are introduced as additional constraints specific for the two proposed placement strategies named WML and WMP.

1) *Constraints associated to the model variables:* in the event that a UAV  $\bar{s}$  is not available (because it has temporarily left the FANET to replace its battery), the Boolean variable defined by (12), associated to such a UAV  $\bar{s}$  and representing whether it is used for placement or not, must be equal to zero:

$$z_{\bar{s}} = 0 \quad \forall \bar{s} \in \tilde{S}, \quad (24)$$

where  $\tilde{S}$  represents the set of unavailable UAVs. From constraint (24) we also obtain  $N_{\bar{s}}^{(VF)} = 0$ , that is equivalent to  $w_{f_k \bar{s}} = 0$ ,  $\forall k$ ,  $\forall f_k$ , and  $x_{i_k \bar{s}} = 0$ ,  $\forall k$ ,  $\forall i_k$ , namely, all the variables related to the UAV  $\bar{s}$  are null.

Furthermore, let us consider, for any UAV  $s$ , for any chain  $k$  and for any function  $f$ , the Boolean variable  $z_s$ . It is easy to prove that the definition in (12) is equivalent to the following constraints:

$$w_{f_k s} \leq z_s \leq 1 \quad \forall f, \forall k, \forall s, \quad (25)$$

$$z_s \leq N_s^{(VF)} \quad \forall s. \quad (26)$$

Indeed, as stated by (25), if at least one function is assigned to the UAV  $s$ , that is, if at least one  $w_{f_k s}$  equals 1, then  $z_s = 1$  as well (the UAV  $s$  is used for placement); on the contrary, if no function is assigned to the UAV  $s$ , that is, if any  $w_{f_k s}$  equals 0 (and therefore their sum,  $N_s^{(VF)}$ , is zero), then  $z_s = 0$ , as established by (26).

Likewise, we have two more constraints regarding the Boolean variable  $z_{fs}$  representing whether the UAV  $s$  is used to run the VF  $f$  or not:

$$w_{f_k s} \leq z_{fs} \leq 1 \quad \forall k, \forall f, \forall s, \quad (27)$$

$$z_{fs} \leq N_{fs}^{(SC)} \quad \forall f, \forall s. \quad (28)$$

Constraints (27) and (28) jointly are equivalent to the definition (13) since, if at least a function  $f$  is assigned to the UAV  $s$

(that is, if at least a  $w_{f_k s} = 1$ ), then  $z_{f_s} = 1$  and, on the contrary, if the sum of all  $w_{f_k s}$  is zero, then  $z_{f_s} = 0$ .

Another constraint related to the model variables regards the condition that each VF instance belonging to a placed chain (for which  $y_k = 1$ ) is assigned to one UAV, otherwise, if a chain is not placed ( $y_k = 0$ ), none of its VF instances has to be allocated to any UAV:

$$\sum_{\forall s \in V} x_{i_k s} = y_k \quad \forall k = 1, \dots, K, \quad \forall i_k = 1, \dots, L_k. \quad (29)$$

In other words, the number of UAVs to which the  $i$ -th VF of the SC  $k$  must be assigned, is equal to the Boolean variable  $y_k$  (that is 1 if the SC  $k$  has been placed, 0 otherwise).

The last constraint associated to the model variables is referred to the Boolean variable  $y_{f_s}$  defined in (15), representing whether the VF  $f$  is shared by at least two SCs on the UAV  $s$  or not. It is easy to prove that the definition (15) is equivalent to the following constraints:

$$y_{f_s} \leq \left( N_{f_s}^{(SC)} - 1 \right) \left( N_{f_s}^{(SC)} \right) \quad \forall f, \forall s, \quad (30)$$

$$\left( N_{f_s}^{(SC)} - 1 \right) \left( N_{f_s}^{(SC)} \right) (1 - y_{f_s}) = 0 \quad \forall f, \forall s. \quad (31)$$

Indeed, as established by (30), if the function  $f$  is not shared on the UAV  $s$  (that is  $N_{f_s}^{(SC)}$  equals 0 or 1), then  $y_{f_s} = 0$ ; on the contrary, if the number of SCs that share the same VF in the same UAV is greater than or equal to 2, then  $y_{f_s} = 1$ , as stated by (31).

Note that, if all the functions appear in at most two different SCs, constraint (31) can be replaced by:

$$y_{f_s} \geq \left( N_{f_s}^{(SC)} - 1 \right) \quad \forall f, \forall s.$$

2) *Constraints associated to the infrastructure limitations:* the first constraint regarding the FANET infrastructure limitations imposes that the aggregate bit rate on each output link of a UAV  $s$  towards each other UAV  $\bar{s}$  is limited by the link transmission rate,  $\bar{T}_{s\bar{s}}$ :

$$\sum_{k=1}^K \sum_{i_k=1}^{L_k-1} x_{i_k s} x_{(i+1)_{k\bar{s}}} \lambda_{i_k}^{(B,IN)} \leq \bar{T}_{s\bar{s}} \quad \forall s, \forall \bar{s}. \quad (32)$$

Note that, since the  $x_{i_k s} x_{(i+1)_{k\bar{s}}}$  product is not null, the above capacity constraint applies only to UAVs  $s$  and  $\bar{s}$  that are subsequent for the chain  $k$  (the UAV  $s$  where the  $i$ -th VF is placed and the UAV  $\bar{s}$  where the  $(i+1)$ -th VF is placed). If two UAVs  $s$  and  $\bar{s}$  are not directly connected with a single-hop link, we set  $\bar{T}_{s\bar{s}} = 0$ . However, this case is not considered in this paper.

The second infrastructure-related constraint imposes that the utilization coefficient of all the queues is less than 1, that is:

$$\psi_{i_k s}^{(P)} < \mu_{i_k s} \quad \forall k, \forall i_k, \forall s. \quad (33)$$

In addition, let us note that constraint (33) guarantees that the overall computation load required by the VF instances placed on the UAV  $s$  is not higher than its computation

capacity,  $\bar{C}_s$ , also representing the maximum flow packet rate that the UAV  $s$  can tolerate:

$$\sum_{k=1}^K \sum_{i_k=1}^{L_k} x_{i_k s} \sigma_{i_k s} \lambda_{i_k}^{(P,IN)} \leq \bar{C}_s \quad \forall s. \quad (34)$$

3) *Constraints associated to the application layer:* the only constraint specified at the application layer imposes that the service response time, defined as the mean end-to-end delay for each chain  $k$ , is not higher than a given threshold,  $\bar{D}_k$ , representing the maximum tolerable delay for that chain:

$$\sum_{i_k=1}^{L_k-1} \sum_{\forall s \in V} x_{i_k s} \left( r_{i_k s} + \sum_{\forall \bar{s} \in V} d_{s\bar{s}} x_{(i+1)_{k\bar{s}}} \right) \leq \bar{D}_k, \quad \forall k, \quad (35)$$

where  $d_{s\bar{s}}$  is the propagation delay on the link  $l_{s\bar{s}}$ , while  $r_{i_k s}$  is the average response time of the  $i$ -th VF in the chain  $k$ , defined as in (7).

4) *Constraints for MPL and WMP strategies:* in Section III, two VNF placement strategies have been introduced to be applied in the proposed framework: the *MPL* and the *WMP*. To better understand how these two strategies work and their differences, they are illustrated in Algorithms 1 and 2, respectively, for the FANET modification event that occurs at the generic time  $t$ .

More in details, two types of events can occur during a FANET mission: a UAV  $\tilde{s}$  leaves the FANET (event  $E_{\tilde{s}}^L$ ) or a UAV  $\tilde{s}$  returns active in the FANET (event  $E_{\tilde{s}}^R$ ). This information (i.e. the UAV  $\tilde{s}$  that causes the event and the type of event) represents the Input of Algorithm 1. Depending of the type of event (line 1), if the event  $E_{\tilde{s}}^L$  occurs, the index  $\tilde{s}$  belongs to the set  $\tilde{S}$  of the unavailable UAVs (lines 2-3) and, hence, the variable  $z_{\tilde{s}}$  defined in (12) is set to 0, as established by constraint (24); otherwise, if the event is  $E_{\tilde{s}}^R$ , the index  $\tilde{s}$  not belongs to the set  $\tilde{S}$  (lines 4-5), and the variable  $z_{\tilde{s}}$  could assume any value. Due to the fact that the MLP strategy performs a memoryless placement not considering the previous position of VFs on UAVs, we set both  $z_{f_s}^{(t)}$  and  $y_{f_s}^{(t)}$  equal to 0,  $\forall f \in F$  and  $\forall s \in V$  (lines 6-7), in order to remove the optimal placement derived at the time  $t-1$ . The new matrices created in this way are respectively saved in  $\hat{Z}^{(t)}$  and  $\hat{Y}^{(t)}$ . These two matrices are used to save the new optimal placement at time  $t$  derived as the solution of the optimization problem, and in particular minimizing the objective function  $\Theta$  subject to constraints (24)-(35), given as Data of the Algorithm 1 (line 8). At the end (line 9), the MLP Algorithm returns the optimal placement solution that has to be used by UAVs composing the FANET until the next FANET modification event.

Referring to Algorithm 2, applied when the WMP strategy is used, also in this case the UAV  $\tilde{s}$  causing the event and the type of event (either  $E_{\tilde{s}}^L$  or  $E_{\tilde{s}}^R$ ) are provided as Input of the Algorithm 2. In addition, due to the fact that optimization is applied to decide the placement of the only VFs that were running on that UAV  $\tilde{s}$ , while the allocation of all the other VFs remains unchanged with respect to the previous placement, also the  $z_{f_s}^{(t-1)}$  and  $y_{f_s}^{(t-1)}$ ,  $\forall f \in F$  and  $\forall s \in V$ , are needed as Input. Furthermore, when the UAV  $\tilde{s}$  comes

---

**Algorithm 1** MLP Algorithm
 

---

**Input:** UAV  $\tilde{s}$ ,  $event_{\tilde{s}} \in \{E_{\tilde{s}}^L, E_{\tilde{s}}^R\}$   
**Output:**  $\hat{Z}^{(t)}, \hat{Y}^{(t)}$   
**Data:**  $\Theta$ , constrains (24)-(35)

- 1: **switch**  $event_{\tilde{s}}$  **do**
- 2:   **case**  $E_{\tilde{s}}^L$  ▷ UAV  $\tilde{s}$  leaves the FANET
- 3:      $\tilde{s} \in \tilde{S} \Rightarrow z_{\tilde{s}} = 0$
- 4:   **case**  $E_{\tilde{s}}^R$  ▷ UAV  $\tilde{s}$  returns active
- 5:      $\tilde{s} \notin \tilde{S}$
- 6:      $\hat{Z}^{(t)} \leftarrow z_{fs}^{(t)} = 0, \forall f \in F, \forall s \in V$
- 7:      $\hat{Y}^{(t)} \leftarrow y_{fs}^{(t)} = 0, \forall f \in F, \forall s \in V$
- 8:      $[\hat{Z}^{(t)}, \hat{Y}^{(t)}]$  from the solution of (23) s.t. (24)-(35)
- 9: **return**  $\hat{Z}^{(t)}, \hat{Y}^{(t)}$

---

back to the FANET, the WMP strategy tries to place inside it the same VFs of the placement in which all the UAVs were active. This placement, represented by the matrix  $\tilde{Z}^{(1)}$ , is given as Input as well.

As for Algorithm 1, according to the type of event (line 1), if the event is  $E_{\tilde{s}}^L$  then  $\tilde{s} \in \tilde{S}$  and, hence, we set  $z_{\tilde{s}}$  equal to 0 (lines 2-3); otherwise,  $\tilde{s} \notin \tilde{S}$  if the event is  $E_{\tilde{s}}^R$  (lines 4-5). In this last case, as said before, the WMP tries to place in the UAV  $\tilde{s}$  the same VFs of the optimal placement with all the  $N$  UAVs active. For this reason, for each function  $i \in F$ , the algorithm checks if  $\tilde{Z}_{(i,\tilde{s})}^{(1)}$  is equal to 1, meaning that the  $i$ -th function could be placed in the UAV  $\tilde{s}$  now (lines 6-7). To this purpose, we set both  $z_{is}^{(t-1)}$  and  $y_{is}^{(t-1)}$  equal to 0  $\forall s \in V$  to force the algorithm to replace only these functions (lines 8-9).

After that,  $z_{fs}^{(t-1)}$  and  $y_{fs}^{(t-1)}$ ,  $\forall f \in F$  and  $\forall s \in V$ , are saved inside the matrices  $\hat{Z}^{(t-1)}$  and  $\hat{Y}^{(t-1)}$  (lines 12-13).

To solve the optimization problem using WMP strategy, it is necessary to introduce two more constraints for each possible type of event. When the UAV  $\tilde{s}$  leaves the FANET, we define:

$$z_{\hat{f}\hat{s}} = 1 \quad \forall \hat{s} \notin \tilde{S}, \forall (\hat{f}, \hat{s}) \in \hat{Z}_1^{(t-1)}, \quad (36)$$

$$y_{\hat{f}\hat{s}} = 1 \quad \forall \hat{s} \notin \tilde{S}, \forall (\hat{f}, \hat{s}) \in \hat{Y}_1^{(t-1)}, \quad (37)$$

In (36),  $\tilde{S}$  is the set of unavailable UAVs (note that  $\tilde{s} \in \tilde{S}$ ) while  $\hat{Z}_1^{(t-1)} = \{(f, s) \in F \times V : z_{fs}^{(t-1)} = 1\}$  is the set of ordered pairs (function, UAV) which had been associated in the previous optimal placement. Likewise, in (37), we define  $\hat{Y}_1^{(t-1)} = \{(f, s) \in F \times V : y_{fs}^{(t-1)} = 1\}$ .

Instead, when a UAV  $\tilde{s}$  comes back to the FANET after battery substitution, we define:

$$z_{\hat{f}\hat{s}} = 1 \quad \forall \hat{s} \notin \tilde{S}, \forall \hat{f} \notin \tilde{F}_{\tilde{s}}^{(1)}, \forall (\hat{f}, \hat{s}) \in \hat{Z}_1^{(t-1)}, \quad (36')$$

$$y_{\hat{f}\hat{s}} = 1 \quad \forall \hat{s} \notin \tilde{S}, \forall \hat{f} \notin \tilde{F}_{\tilde{s}}^{(1)}, \forall (\hat{f}, \hat{s}) \in \hat{Y}_1^{(t-1)}. \quad (37')$$

where  $\tilde{F}_{\tilde{s}}^{(1)} = \{f \in F : z_{f\tilde{s}}^{(1)} = 1\}$  is the set of the VFs allocated on UAV  $\tilde{s}$  at the first placement (that is, when  $\tilde{S} = \emptyset$ ).

All that said, returning to the description of the Algorithm 2, we consider two matrices,  $\hat{Z}^{(t)}$  and  $\hat{Y}^{(t)}$ , used to save the

new VFs placement at time ( $t$ ). Depending on the type of event (line 14), the VFs placement is derived minimizing the objective function  $\Theta$  subject to:

- constraints (24)-(37) if event is equal to  $E_{\tilde{s}}^L$  (lines 15-16);
- constraints (24)-(35), (36')-(37') if event is equal to  $E_{\tilde{s}}^R$  (lines 17-18).

At the end (line 19), the WMP Algorithm returns the optimal placement solution that has to be used by UAVs composing the FANET until the next event.

---

**Algorithm 2** WMP Algorithm
 

---

**Input:** UAV  $\tilde{s}$ ,  $event_{\tilde{s}} \in \{E_{\tilde{s}}^L, E_{\tilde{s}}^R\}, z_{fs}^{(t-1)}, y_{fs}^{(t-1)}, \forall f \in F, \forall s \in V, \hat{Z}^{(1)}$   
**Output:**  $\hat{Z}^{(t)}, \hat{Y}^{(t)}$   
**Data:**  $\Theta$ , constrains (24)-(35),(36),(37),(36'),(37')

- 1: **switch**  $event_{\tilde{s}}$  **do**
- 2:   **case**  $E_{\tilde{s}}^L$  ▷ UAV  $\tilde{s}$  leaves FANET
- 3:      $\tilde{s} \in \tilde{S} \Rightarrow z_{\tilde{s}} = 0$
- 4:   **case**  $E_{\tilde{s}}^R$  ▷ UAV  $\tilde{s}$  returns active
- 5:      $\tilde{s} \notin \tilde{S}$
- 6:     **for**  $i = 1 : |F|$  **do**
- 7:       **if**  $\hat{Z}_{(i,\tilde{s})}^{(1)} == 1$  **then**
- 8:          $z_{is}^{(t-1)} = 0, \forall s \in V$
- 9:          $y_{is}^{(t-1)} = 0, \forall s \in V$
- 10:       **end if**
- 11:     **end for**
- 12:      $\hat{Z}^{(t-1)} \leftarrow z_{fs}^{(t-1)}, \forall f \in F, \forall s \in V$
- 13:      $\hat{Y}^{(t-1)} \leftarrow y_{fs}^{(t-1)}, \forall f \in F, \forall s \in V$
- 14:   **switch**  $event_{\tilde{s}}$  **do**
- 15:     **case**  $E_{\tilde{s}}^L$
- 16:        $[\hat{Z}^{(t)}, \hat{Y}^{(t)}]$  from the solution of (23) s.t.(24)-(37)
- 17:     **case**  $E_{\tilde{s}}^R$
- 18:        $[\hat{Z}^{(t)}, \hat{Y}^{(t)}]$  from the solution of (23) s.t.(24)-(35), (36') and (37')
- 19: **return**  $\hat{Z}^{(t)}, \hat{Y}^{(t)}$

---

## VI. NUMERICAL RESULTS

In this section, we will consider a use case to evaluate the performance of the proposed solution. More specifically, in Section VI-A we describe the use case setup. Then, in Section VI-B, we discuss about model complexity and resolution times. Section VI-C presents an extensive simulation campaign to discuss about the impact of some system parameters on the performance of the FANET and the Quality of Service (QoS) provided to its users, and to compare the proposed framework with a standard solution that is taken as reference to demonstrate the achieved gain. Finally, Section VI-D compares MLP and WMP to evaluate performance worsening occurring when we apply WMP to decrease flow re-routing after FANET modification events.

### A. Use Case Setup

In this section, we describe the scenario we consider as use case for performance evaluation. The main parameters are

TABLE IV: Use case setup

Parameter	Value
$ V $	5
$K$	5
$ F $	8
$L_k$	$3 \forall k \in K$
$p_s^{(E)}$	[66.56 66.56 66.56 8 66.56] W
$C_s$	$8.32 \cdot 10^9$ op/s $\forall s \in V$
$\lambda_k^{(B)}$	[119.4 255.798 111.325 132.401 48.83] Mbps
$\Omega$	0.95375 kbyte
$T_{s\bar{s}}$	1 Gbps, $\forall s, \bar{s} \in E$ , with $s \neq \bar{s}$
$d_{s\bar{s}}$	$\sim 0$ s, $\forall s, \bar{s} \in E$ , with $s \neq \bar{s}$
$D_k$	1 ms, $\forall k \in K$
$\alpha_P$	0.5
$\alpha_G$	0.1

TABLE V: Packet-rate and bit-rate input/output ratios

VF	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
$\gamma_f^{(P)}$	0.5	0.8	1	1	0.7	1	1	1
$\gamma_f^{(B)}$	0.5	0.9	1.2	1	0.7	1.7	1	1.8

summarized in Table IV. Let us consider a FANET of  $|V| = 5$  UAVs that has to provide devices on ground with  $K = 5$  SCs, with length  $L_k = 3, \forall k \in [1, K]$ . The required SCs are described in Fig. 3. They chain the following  $|F| = 8$  VFs:

- $f_1$ : Next-generation firewall (NGFW)
- $f_2$ : Secure Email Gateway (SEG)
- $f_3$ : Internet Protocol Security (IPsec)
- $f_4$ : Virtual Private Network (VPN)
- $f_5$ : Web Application Firewall (WAF)
- $f_6$ : Augmented Reality Composer (ARC)
- $f_7$ : Video Flow Classifier (VFC)
- $f_8$ : Secure Sockets Layer (SSL)

The bit rates  $\lambda_k^{(B)}$  of the five aggregated flows using the FANET, as well as the packet-rate and bit-rate input/output ratios,  $\gamma_f^{(P)}$  and  $\gamma_f^{(B)}$ , characterizing each function  $f \in F$  were estimated on a deployment at the *UniCT 5G&B Lab*. They are listed in Tables IV and V, respectively. As we can see in Table V, for example the function  $f_1$  halves both packet rate and bit rate, being a firewall, while  $f_6$  increases the bit rate only, performing enrichment of video flow packets. In the same testbed we have also measured the mean packet size,  $\Omega = 0.95375$  kbyte.

In order to reproduce a use case as much closer to the reality as possible, we considered that each VF has different implementations. Specifically, as listed in Table VI, all VFs have two implementations, here referred to as *Big Instances* and *Small Instances*. Moreover, VFs  $f_4, f_5$  and  $f_6$  have an additional extra-light implementation, named *Tiny Instance*. These implementations provide the same service, but differ from each other for their performance and, on the other hand, for their complexities, the latter expressed in number of Floating Operations Per Packet (FLOPP).

UAVs have installed, by default, the Big Instance of all the functions, except the following cases:

- UAV  $s_1$  has the Small Instance of  $f_5, f_6, f_7$  and  $f_8$ ;

TABLE VI: VF instance complexity

VF Instance complexity (FLOPP)			
VF	Big	Small	Tiny
$f_1$	53280	9411	//
$f_2$	39664	35277	//
$f_3$	72852	28983	//
$f_4$	49580	27645	5711
$f_5$	44622	27074	753
$f_6$	49408	27474	5539
$f_7$	66106	22238	//
$f_8$	71395	27526	//

TABLE VII:  $p_{fs}^{(VM)}$  for each VF

$p_{fs}^{(VM)} \forall s \in V$ (Watt)							
$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
8.63	10.78	1.22	15.41	20.01	3.69	1.22	2.78

- UAV  $s_2$  has the Tiny Instance of  $f_4, f_5$  and  $f_6$ ;
- UAV  $s_3$  has the Small Instance of  $f_3, f_4$  and  $f_7$ ;
- UAV  $s_4$  has the Small Instance of  $f_2$ ;
- UAV  $s_5$  has the Small Instance of  $f_1$ .

All UAVs are equipped with the same CE that is able to execute 8.32 gigaFLOPS/s.

Concerning power consumption for engines, we consider four small equal quadcopters consuming a power of 67 W, and one low-power UAV, specifically a balloon, referred to as  $s_4$ , consuming only 8 W, only to maintain its position. The values of power consumption to maintain the VMs hosting the VFs switched on are listed in Table VII, while the energy needed to process a packet for the VMs hosting the VF  $f$  in the UAV  $s$ , i.e.  $e_{fs}^{(P)}$ , can be derived from the VF instance complexities  $\sigma_{fs}$  listed in Table VII considering that each floating point operation consumes 132 nJ to be executed in the considered CE.

Concerning the wireless links between UAVs, considering that all UAVs are not too far away from each other and are all in Line Of Sight (LOS), we assume the same average transmission rate of 1 Gbit/s, equal for each link, an average power consumption for transmission of 6W and a negligible propagation delay.

Finally, we consider that all the chains have the same requirements in terms of maximum tolerable delay, equal to 1 ms [63], referred to the end-to-end delay in the FANET, as expressed by (35). The two weights of the objective function  $\Theta$  are set as  $\alpha_P = 0.5$  for the power consumption contribution and  $\alpha_G = 0.1$  for the "Satisfied Flows" gain contribution.

Numerical results are evaluated by a Matlab-based event-driven simulator that is able to reproduce the flight duration of the UAVs, considering the power consumption due to all the terms specified in Section V-B, the battery capacity,  $B$ , mounted on board the UAVs, and the time,  $t_{BR}$  needed for each UAV to reach the Charging Station, replace its empty battery with a charged one, and come back to the FANET. These two last parameters have been varied in the simulation campaign.

The simulator, at each event of UAV landing and UAV

rejoining the FANET, accesses a database to retrieve the results of the optimization model to find the optimal placement. This database, for both cases of WMP and MLP strategies, was filled offline by running the optimization model described so far. Performance of the optimization model will be described in Section VI-B.

### B. Model Evaluation

The proposed model represents a nonlinear multi-objective optimization problem; indeed the model involves more than one objective function which must be minimized. We used the weighted sum method that combines and converts all the objective functions into a single-objective composite function using the weighted sum (see [64], [65] and [66]). Furthermore, since some of the terms of the objective function are nonlinear and the feasible region is determined by some nonlinear constraints, we are dealing with a nonlinear programming problem in which the variables are Boolean.

We solved the problem using CP Optimizer, the CPLEX Constraint Programming solver, which rapidly solves a large range of problems traversing and systematically exploring the decision tree for feasible and efficient solutions (through domain reduction) [67]. The CPLEX solver presolves the input model before search, modifying the model to improve it so that it can be solved more efficiently. Indeed, CP Optimizer automatically removes redundant constraints, reformulates the model to use constraints that are more efficiently propagated and identifies conflicting constraints (if any).

We obtained solutions using the CPLEX Solver on a laptop with an HP 255.5 computer, 5 compute cores 2C+3G, 2.60 GHz, RAM: 8 GB. For the placement of the SCs described in Fig. 3, the MLP Model previously described was repeatedly executed for each configuration representing each FANET modification event or change of network topology. We observe that the number of variables is constant for each configuration (this allows us to better manage the CPLEX call for each different configuration): the problem has 440 variables. Instead, the number of constraints constituting the model decreases in the events when some UAVs are not available in the FANET (constraint (24) prevents their use). Therefore, the model is constituted by 950 constraints for each configuration with only one UAV available, 949 constraints for each configuration with two UAVs available, 948 for three UAVs available, 947 for each configuration with four UAVs available and 948 constraints for the configuration with all the UAVs available. The values reported in Fig. 5 refer to the configurations which have been divided according to the number of active UAVs to be used for placement (ranging between 1 and 5). More specifically, Fig. 5 represents the averages of total memory usage, required computation time and number of feasible solutions found for all configurations with 1, 2, 3, 4 or 5 UAVs available for placement, respectively.

By paying particular attention to the computational time required to perform the model and to compute the solutions of each configuration, we note that it belongs to the interval [0.31, 40.32]s, the arithmetic mean of the time spent to solve

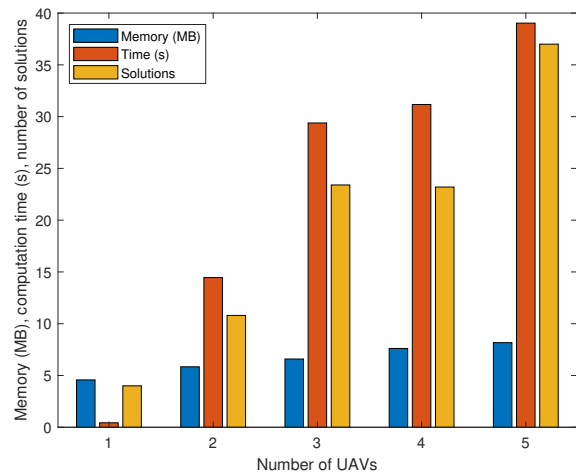


Fig. 5: Memory usage, computation time and number of feasible solutions

is 20.5s; therefore the solutions, although calculated offline, are obtained in a fairly short time.

### C. Comparison with a Standard Placement Technique

As already pointed out in Sections I and II, the main novelties introduced in this paper, as compared with the previous literature, regard both the system, i.e. a FANET to provide ground devices with a softwarized network equipped with edge computing facilities, and the placement strategy, which is able to minimize power consumption and maximize the total packet rate managed by the FANET. Starting from the consideration that, at the best of our knowledge, there is no previous work with both these objectives, here we compare the MLP strategy proposed in this paper with a standard placement model that we will name *No-share placement (NoShP)*. It is memoryless as MLP, but does not share VFs among different flows and does not take into account the packet-rate and bit-rate input/output ratios  $\gamma_f^{(P)}$  and  $\gamma_f^{(B)}$ . Therefore, referring to Fig. 3, it tries to deploy fifteen instances of the VFs needed to create the five required chains, all with  $\gamma_f^{(P)} = 1$  and  $\gamma_f^{(B)} = 1$ .

First, we will analyze how the FANET behavior is influenced by its main parameters, i.e. the round-trip time needed by a UAV to replace the battery,  $t_{BR}$ , and the battery capacity,  $B$ .

Figs. 6, 7, 8, and 9 are correlated to each other. In all these figures, we can observe two different behaviors, one for low values of  $t_{BR}$  and one for high values of it; the boundary between the two intervals depends on the battery capacity  $B$ . In the left interval, the average number of VF instances deployed on the same UAV increases with  $t_{BR}$  (see Fig. 6) because the same VFs have to be placed on less UAVs while the number of active UAVs decreases, as we can note in Fig. 7. For this reason, as shown in Fig. 8, the average power consumption of each active UAV increases and, consequently, the average UAV flight time, shown in Fig. 9, decreases. Instead, in the right interval of these figures, which

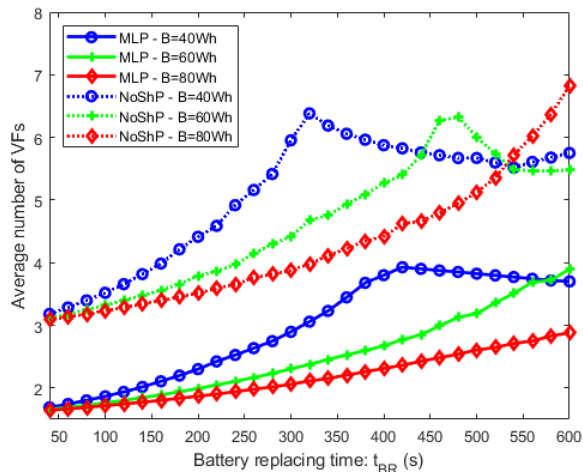


Fig. 6: Average number of VFs running on each UAV (MLP vs. NoShP)

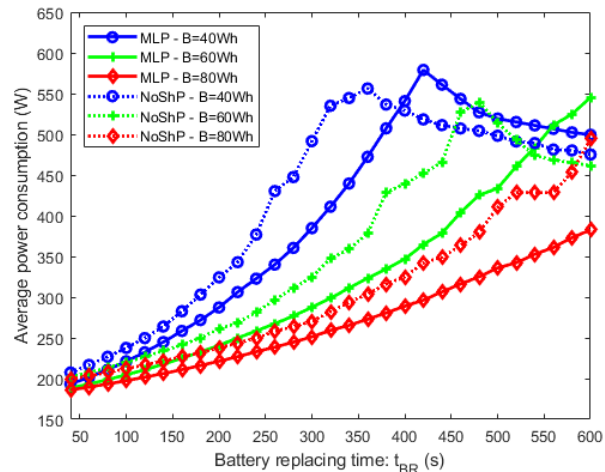


Fig. 8: Average power consumption of each active UAV (MLP vs. NoShP)

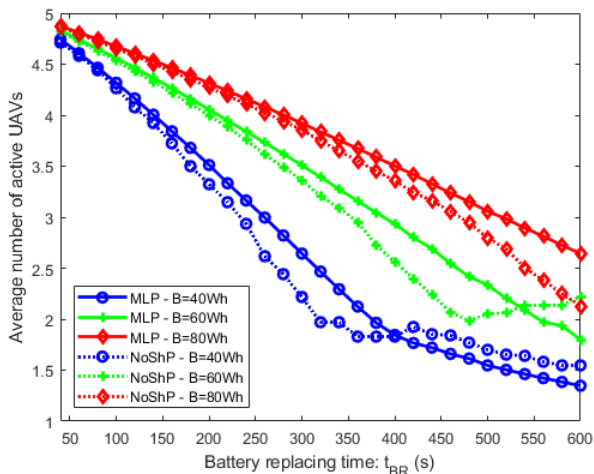


Fig. 7: Average number of flying UAVs (MLP vs. NoShP)

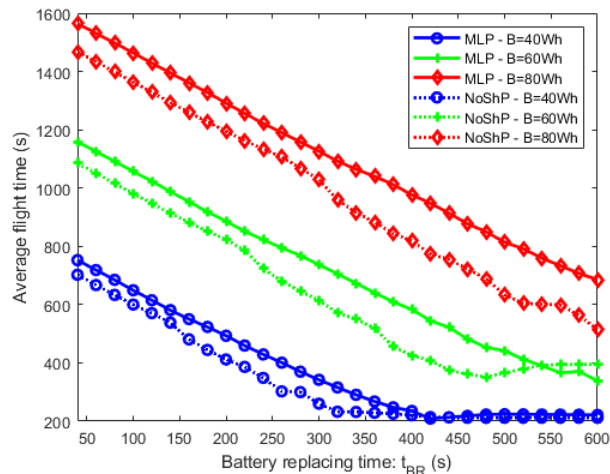


Fig. 9: Average UAV flight time (MLP vs. NoShP)

is characterized by a FANET with a too low number of active UAVs, as shown in Fig. 7, active UAVs are in saturation, and the average number of deployed VFs decreases because the few active UAVs are no longer sufficient to accommodate all the chains, and therefore, some VFs are not deployed. For the same reason, the average power consumption slowly decreases until becoming almost constant, as evident in Fig. 8. This occurs because, on average, for very high round-trip times to/from the Charging Station, the lower power consumption for a UAV when it is active in the FANET (which is decreased because some SCs are not deployed for lack of FANET resources) is compensated by the increment of power consumption needed to reach the farther Charging Station. The same behavior influences, of course, the average flight time.

As far as the comparison between the two placement models, MLP and NoShP, the four above figures demonstrate the

gain achieved with the proposed framework. More specifically, Fig. 6 confirms that NoShP imposes a higher number of VFs to be deployed on each UAV for any value of the battery replacing time,  $t_{BR}$ . This causes that the right interval (the interval characterized by UAV saturation) is larger than in the case of MLP.

Also, we can observe that, in the left interval of Figs. 8, 9 and 7, NoShP has a higher average power consumption and, consequently, a lower average flight time and a lower number of active UAVs. On the contrary, in the right interval of the above figures, we have an inversion of the two curves for each case of  $B$  (it is more evident for  $B = 40$  Wh) with respect to the left interval. This occurs because, in this case, due to the higher level of network saturation related to the longer battery replacing time, the "Satisfied Flows" gain contribution becomes dominant over the power consumption

in the objective function. Therefore, the optimization process privileges the placement of the SCs that allow to maximize the served packet rate over the one that allows more energy saving. This is the reason why the WMP strategy shows an average power consumption higher than NoShP (see Fig. 8). This causes a consequent lower number of UAVs available (see Fig. 7) because we have only those that are able to maximize the number of satisfied flows. As observed so far, the average number of VFs that are deployed on each active UAV, shown in Fig. 6, remains higher for NoShP because this strategy, not sharing VFs among different flows, is forced to maintain a higher number of VFs to serve the same flows, but this is critical in saturation conditions. Consequently, fewer chains are placed with NoShP, and therefore fewer VF instances are run, with a consequent reduction of power consumption.

In order to further analyze how the placement optimization model works, in Fig. 10 we show the average value of the objective function for both MLP and NoShP. This function has been measured for each number of active UAVs because both the terms in (22), i.e. power consumption and gain achieved by serving SFs, depend on it. In the overall figure (without considering the magnified part of it), we can appreciate how much the objective function is dominated by the gain achieved by serving satisfied flows,  $G_{SF}$ , when not all the chains are placed (mainly during periods in which only one UAV is active in the FANET). The high values characterizing the objective function in this case demonstrate how much bad is considered it by the FANET Orchestrator. On the contrary, when at least two UAVs are active in the FANET, the FANET Orchestrator almost always is able to place all the chains, so achieving very low values of the objective function. In all these cases, the second term of the objective function is maximum and equal for all of them, and therefore all values are below  $-8000$ . Comparison between these cases, only dependent on the first part of the objective function (i.e. the part that considers power consumption), is shown in the magnified part of the figure. In this subfigure, not only we can observe that, as expected, the minimum value of the objective function improves (i.e. decreases) with the number of active UAVs, but also it is better (i.e. lower) when the MLP strategy is applied.

Finally, Figs. 11, 12 and 13 present a comparison of MLP with NoShP in terms of the main performance parameters, i.e. the average delay, the packet loss rate and the service unsatisfaction probabilities. More specifically, the service unsatisfaction probability is expressed in terms of *no-service probability* and *partial-service probability*, respectively which are defined as follows:

$$\varphi_{NS} = \frac{T_{NS}}{T_{TOT}} \quad \varphi_{PS} = \frac{T_{PS}}{T_{TOT}} \quad (38)$$

where  $T_{TOT}$  is the duration of a simulation run,  $T_{NS}$  is the sum of the time duration when no SCs have been deployed in the FANET, so no flows have been served, and  $T_{PS}$  is the sum of the time duration when not all the required SCs have been deployed.

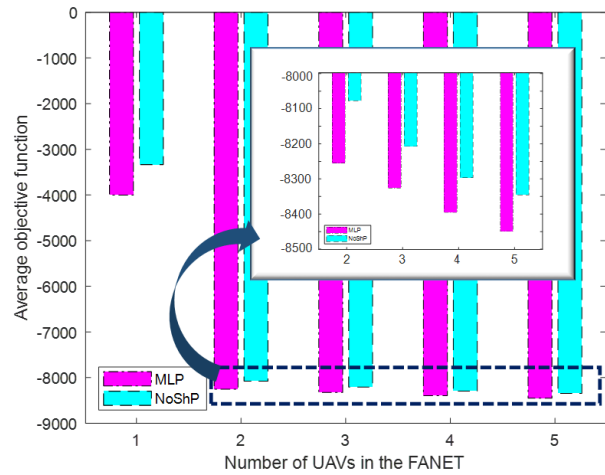


Fig. 10: Objective function (MLP vs. NoShP). Magnification:  $x$ -axis:[2, 5] and  $y$ -axis:[ $-8500, -8000$ ]

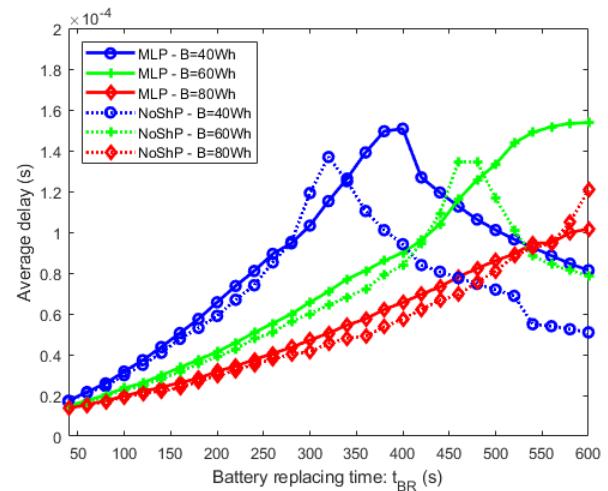


Fig. 11: Average delay suffered in the FANET (MLP vs. NoShP)

Observe that all the above mentioned "quantitative" metrics give an indication of the service continuity which represents a "qualitative" performance metric and expresses how well our design is able to satisfy service requests and support users everytime, in a continuous and uninterrupted way. The impact of these parameters associated to service continuity for ground devices strongly depends on the vertical application and possible alternative solutions that can be implemented. For example, if a vertical application is not able to tolerate packet loss, an alternative solution may consist in setting up a backup connection to a remote cloud, for example via a satellite link. However, this solution can result very expensive and could not guarantee the required delay performance. So it is a matter of identifying a design trade-off.

The non-monotonic behavior of the average delay, shown in Fig. 11, is motivated by the same reasoning discussed so



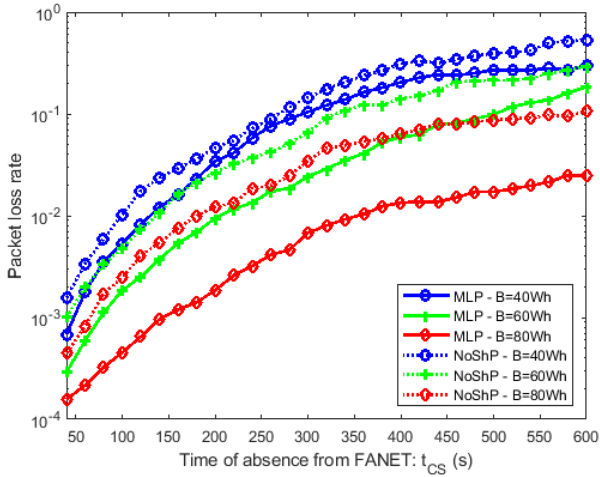
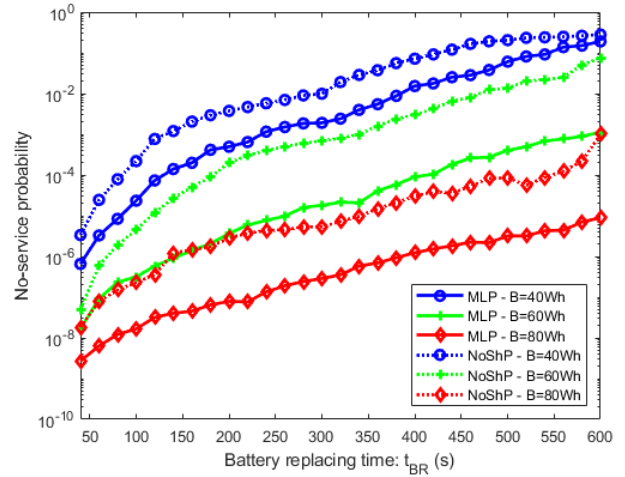
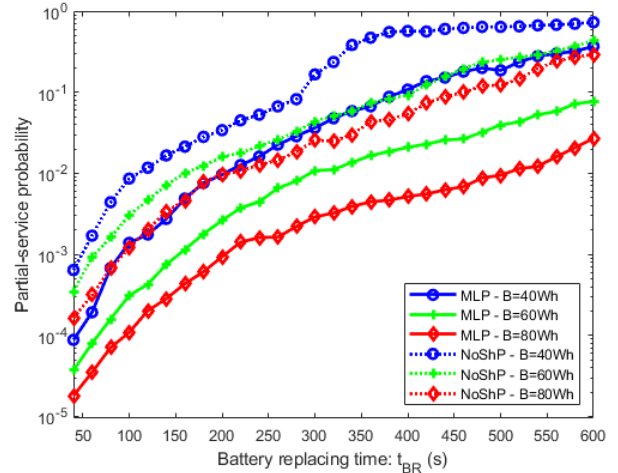


Fig. 12: Packet loss rate (MLP vs. NoShP)



(a) No-service probability



(b) Partial-service probability

Fig. 13: Service unsatisfaction probabilities (MLP vs. NoShP)

far, with an inversion in the right part of the curves. Instead, the other performance parameters get monotonically worsen when  $t_{BR}$  increases and  $B$  decreases since these conditions increase UAV saturation. In the left part of the average delay curves, the proposed framework coincides with the NoShP, since the number of active UAVs is sufficient to provide the required resources in both the cases. In the middle part, the proposed framework performs better (i.e. presents a lower average delay), while in the right part the average delay measured with the NoShP strategy looks better. Nevertheless, let us notice that the advantage of NoShP in terms of average delay in the right part of Fig. 11 is only ostensible because the achieved values of the average delay remain thoroughly below the imposed constraint of 1 ms for both the strategies. Nevertheless, MLP achieves better performance in terms of packet loss ratio and, more important, it presents a better service continuity having lower no-service and partial-service probabilities (Figs. 12 and 13). In other words, the presented results show that MLP outperforms NoShP while respecting delay constraints.

#### D. Memoryless vs. With-Memory Placement

In the previous section, we have evaluated performance of the MLP strategy, which has been defined with the only target of optimizing performance at each event that modifies the number of active UAVs in the FANET. However, given that it does not take care of the position of VFs immediately before each placement optimization, VFs can be moved from one UAV to another one at each optimization action, with the possibility of re-routing of a high number of flows at each event. This is the reason why we have also introduced the WMP strategy, for its peculiarity of maintaining memory of the previous VF allocation in the FANET.

The main differences between MLP and WMP are evident in terms of average power consumption, as shown in Fig. 14, because optimization with WMP is worse. Also in this case,

we can notice different behaviors. After an interval where the two strategies perform in a similar way, there is an interval where MLP saves a lot of power. Then, there is an interval where WMP consumes less power; this is because there are less active UAVs (see Fig. 15) to support the same number of VFs, but the overall power consumption due to the UAV engines is less. Finally, in the last interval on the right, the FANET is stressed almost at the same way, independently of the used strategy (although this is still not visible for the two curves in red, calculated for  $B = 80$  Wh), tending to the same average number of active UAVs in the FANET, so presenting the same number of running VFs on each UAV (see Fig. 16), and the same average power consumption.

Consequently, the overall behavior of the FANET is worse with WMP, both in terms of packet loss ratio, shown in Fig. 17, and service unsatisfaction probabilities and average delay, shown in Figs. 18 and 19, respectively.

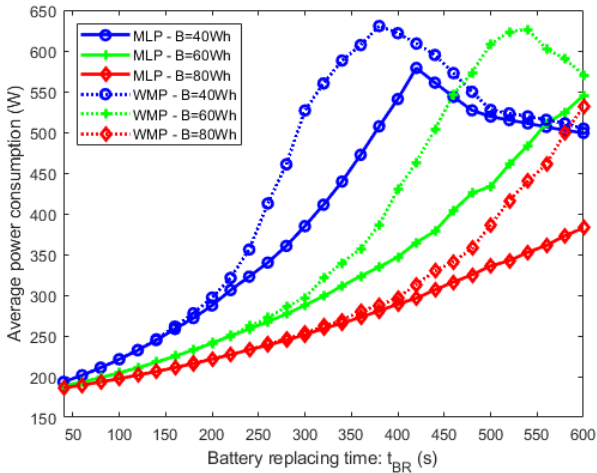


Fig. 14: Average power consumption of active UAVs (MLP vs. WMP)

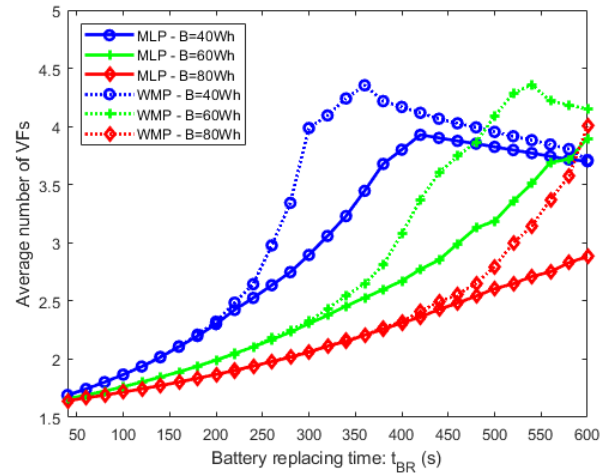


Fig. 16: Average number of VFs running on each UAV (MLP vs. WMP)

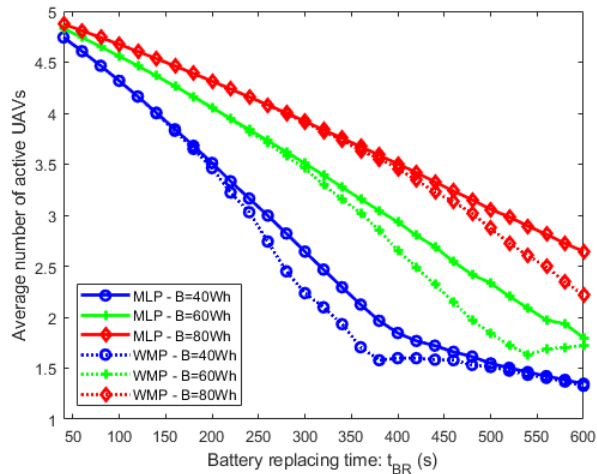


Fig. 15: Average number of flying UAVs (MLP vs. WMP)

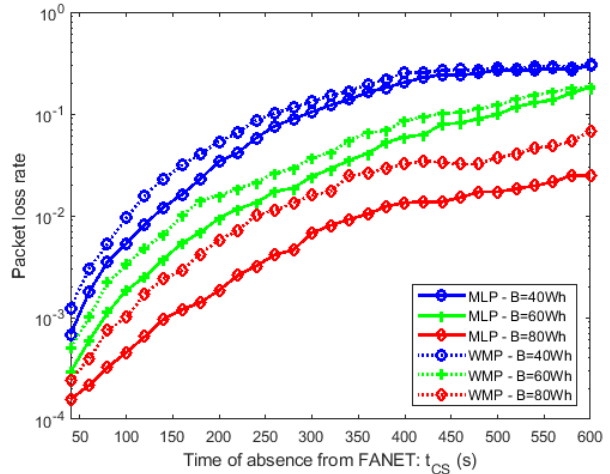


Fig. 17: Packet loss rate (MLP vs. WMP)

However, as desired, we obtain a reduction of the flow re-routing rate, as shown in Fig. 20, and this reduction is higher when the FANET is stressed, i.e. for higher values of the round-trip time needed to replace the battery,  $t_{BR}$ , and for lower values of the battery capacity,  $B$ .

Nevertheless, let us consider that the final choice between MLP and WMP depends on the specific application scenario. Indeed, as known, flow re-routing can cause an abrupt variation of delay for some packets and, as a consequence, possible packet disordering [69] [70]. Moreover, for networks that are based on the SDN paradigm [71], like the FANET considered in this paper, each flow re-routing is realized by a communication between the SDN Controller running in the FANET Coordinator and the SDN Switches running in the Management VF inside each UAV. This causes a transmission overhead and an additional delay that can violate the delay

requirement imposed by the delay threshold described in (35). Evaluation of this impact on performance perceived by the end users, also taking into account the sensitivity of specific vertical applications to possible delay violations that can occur in the timescale of FANET event modifications, is out of scope of this paper. Summarizing, if the above issues caused by re-routing may be a problem for the considered scenario, WMP would be preferred. Otherwise, the strategy that presents better performance is definitely MLP.

## VII. CONCLUSIONS

In this paper we have proposed to employ FANETs to extend the capabilities of 5G networks so densifying and augmenting network coverage in remote areas or where communication infrastructures are not available. This empowering of the 5G architecture to support complex tasks and

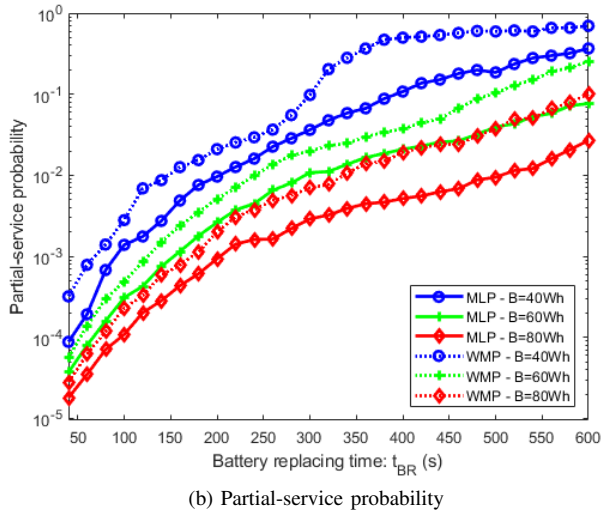
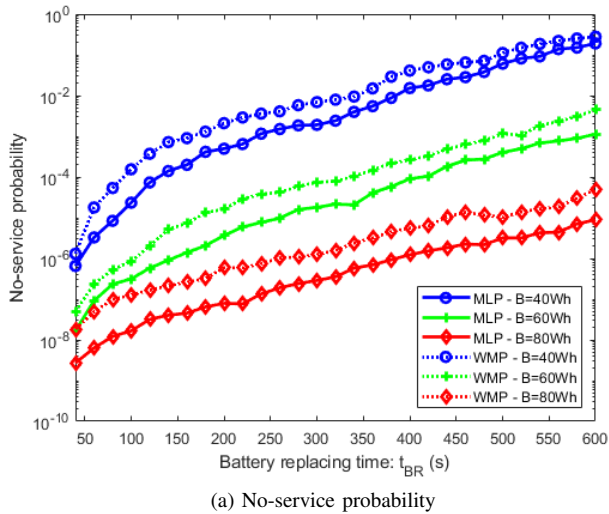


Fig. 18: Service unsatisfaction probabilities (MLP vs. WMP)

allow exploitation of network and application functions made available from the FANET UAVs is implemented through a service chaining mechanism. To take into account the limited energy and computation capabilities of the UAVs, as well as the application constraints in terms of delay and network availability, we have developed a complete and comprehensive optimization framework which does not use heuristics for the solution and takes into account physical constraints of the UAVs, as well as traffic flow features and requirements. We have compared different placement models with and without memory, illustrating the benefits of a memoryless solution in terms of reduction in the number of virtual functions, decreasing in power consumption, improvement in network lifetime and reduction of service unsatisfaction probabilities. In future research it would be interesting to measure how the MLP and WMP strategies, compared with the NoShP standard strategy, affect the end user on the ground that is consuming

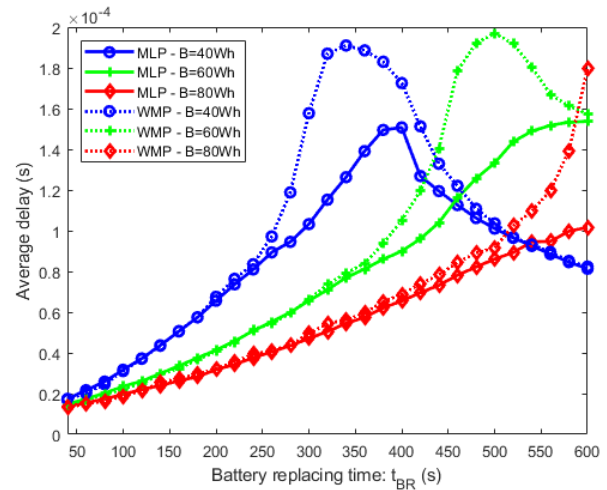


Fig. 19: Average delay suffered in the FANET (MLP vs. WMP)

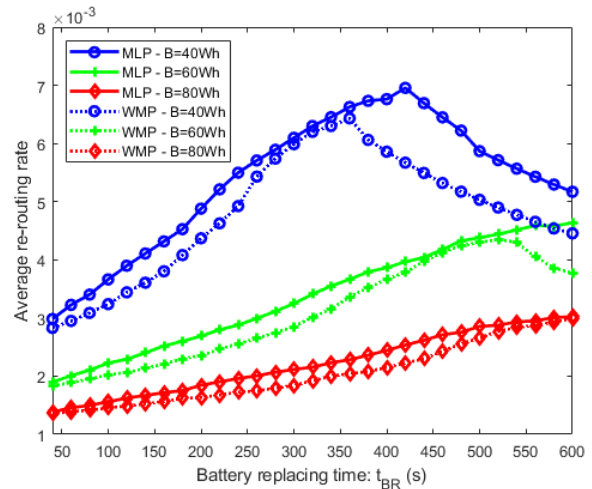


Fig. 20: Average re-routing rate (MLP vs. WMP)

the services through the VFs of the SCs placed on UAVs in the FANET, also analyzing the impact of re-routing, mainly evident with MLP, and the implication of some measured KPIs (round-trip time needed by a UAV to replace the battery, and the battery capacity).

#### ACKNOWLEDGEMENT

The research was partially supported by the research project “Programma ricerca di Ateneo UNICT 2020-22 linea 2-OMNIA” of the University of Catania, the MIUR PRIN project “Liquid\_Edge”, and the POR S6 project.

#### REFERENCES

- [1] I. Bekmezci, O. K. Sahingoz, and S. Temel. Flying Ad-Hoc Networks (FANETs): A survey *Elsevier Ad Hoc Networks* Vol. 11, No. 3. May 2013, Pages 1254-1270.

- [2] M. A. Khan, A. Safi, I. M. Qureshi, I. U. Khan. Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols. *IEEE INTELLECT*, 2017.
- [3] N. Nikhil, S. M. Shreyas, G. Vyshnavi and S. Yadav, "Unmanned Aerial Vehicles (UAV) in Disaster Management Applications," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 140-148.
- [4] A. Saif, K. Dimiyati, K. A. Noordin, N. S. M. Shah, Q. Abdullah and F. Mukhlif, "Unmanned Aerial Vehicles for Post-Disaster Communication Networks," 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), 2020, pp. 273-277.
- [5] M. Peer, V. A. Bohara and A. Srivastava, "Multi-UAV Placement Strategy for Disaster-Resilient Communication Network," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), 2020, pp. 1-7.
- [6] G. Faraci, C. Grasso, G. Schembra. Fog in the Clouds: UAVs to Provide Edge Computing to IoT Devices ACM Transactions on Internet Technology, Volume 20, October 2020
- [7] G. Faraci et al., "Design of a 5G Network Slice Extension With MEC UAVs Managed With Reinforcement Learning," in IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2356-2371, Oct. 2020
- [8] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage," in IEEE Communications Letters, vol. 20, no. 8, pp. 1647-1650, Aug. 2016
- [9] A. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," in IEEE Access, vol. 3, pp. 1206-1232, 2015.
- [10] View on 5G Architecture, 5G PPP Architecture Working Group, Version 3.0, February 2020, Available at: [https://5g-ppp.eu/wp-content/uploads/2020/02/5G-PPP-5G-Architecture-White-Paper\\_final.pdf](https://5g-ppp.eu/wp-content/uploads/2020/02/5G-PPP-5G-Architecture-White-Paper_final.pdf)
- [11] B. Li, Z. Fei, Y. Zhang, and M. Guizani. Secure UAV Communication Networks over 5G. *IEEE Wireless Communications*. Vol. 26, No. 5, October 2019.
- [12] Z. Qadira, F. Ullah, H. S. Munawar, and F. Al-Turjman. Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Elsevier Computer Communications*. Vol. 168, February 2021.
- [13] M. Marchese, A. Moheddine, and F. Patrone. IoT and UAV Integration in 5G Hybrid Terrestrial-Satellite Networks. *MDPI Sensors*. Vol. 19, No. 17, August 2019.
- [14] M. A. Khan, A. Safi, I. M. Qureshi, I. U. Khan. Flying Ad-Hoc Networks (FANETs): A Review of Communication Architectures, and Routing Protocols. *2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, 1-9 - November 2017
- [15] Y. Zeng, R. Zhang, and T. Joon Lim. Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges. *IEEE Communications Magazine*, May 2016
- [16] M. Erdelj, E. Natalizio, K. R. Chowdhury and I. F. Akyildiz, Help from the sky: Leveraging UAVs for disaster management, *IEEE Pervasive Computing*, Vol. 16, NO. 1, March 2017
- [17] ANCHORS project. <http://anchorsproject.org/index.php/en/home/14-das-projekt-i-m-ueherhlick17-projectoverview.html>!
- [18] J. George, P. B. Sujit and J. Sousa, Search strategies for multiple UAV search and destroy missions, *Journal of Intelligent and Robotics Systems* 61 (2011), pp. 355-367.
- [19] Z. Sun, P. Wang, M. C. Vuran, M. Al-Rodhaan, A. Al-Dhelaan and I. F. Akyildiz, BorderSense: border patrol through advanced wireless sensor networks, *Ad Hoc Networks* 9 (3) (2011), pp. 468-477.
- [20] C. Barrado, R. Messeguer, J. Lopez, E. Pastor, E. Santamaria and P. Royo, Wildfire monitoring using a mixed air-ground mobile network, *IEEE Pervasive Computing* 9 (4) (2010), pp. 24-32.
- [21] E. P. de Freitas, T. Heimfarth, I. F. Netto, C. E. Lino, C. E. Pereira, A. M. Ferreira, F. R. Wagner and T. Larsson, UAV relay network to support WSN connectivity, *ICUMT, IEEE*, 2010, pp. 309-314.
- [22] F. Jiang and A. L. Swindlehurst, Dynamic UAV relay positioning for the ground-to-air uplink, *IEEE Globecom Workshops*, 2010.
- [23] A. Cho, J. Kim, S. Lee and C. Kee, Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and pitot tube, *IEEE Transactions on Aerospace and Electronic Systems* 47 (2011), pp. 109-117.
- [24] I. Maza, F. Caballero, J. Capitan, J. R. Martinez-De-Dios and A. Ollero, Experimental results in multi-UAV coordination for disaster management and civil security applications, *Journal of Intelligent and Robotics Systems* 61 (1-4) (2011), pp. 563-585.
- [25] H. Xiang and L. Tian, Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle, *Biosystems Engineering* 108 (2) (2011), pp. 174-190.
- [26] E. Semsch, M. Jakob, D. Pavlicek and M. Pechoucek, Autonomous UAV Surveillance in Complex Urban Environments, *Web Intelligence*, 2009, pp. 82-85.
- [27] Y. Liu, H.-N. Dai, Q. Wang, M. K. Shukla, M. Imran, Unmanned aerial vehicle for internet of everything: Opportunities and challenges, *Elsevier Computer Communications*, Vol. 155, 2020, 10
- [28] C. Han, J. Yin, L. Ye, and Y. Yan "NCant: A Network Coding-Based Multipath Data Transmission Scheme for Multi-UAV Formation Flying Networks", in IEEE Communications Letters, VOL. 25, NO. 3, March 2021.
- [29] S. Rosati, K. Kruzelecki, G. Heitz, D. Floreano and B. Rimoldi, Dynamic Routing for Flying Ad Hoc Networks, in IEEE Transactions on Vehicular Technology, vol. 65, no. 3, March 2016.
- [30] Z. Zheng, A. K. Sangaiah, and T. Wang Adaptive Communication Protocols in Flying Ad Hoc Network IEEE Communications Magazine, January 2018
- [31] K. Daniel and C. Wietfeld, Using public network infrastructures for UAV remote sensing in civilian security operations, Defense Technical Information Center (DTIC) document, Technical University of Dortmund, Germany, Mar. 2011.
- [32] Z. Han, A. L. Swindlehurst, and K. Liu, Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3533-3546, Dec. 2009.
- [33] S. Rohde and C. Wietfeld, Interference aware position in go faerial relays for cell overload and outage compensation, in Proc. IEEE Veh. Technol. Conf. (VTC),
- [34] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. Unmanned Aerial Vehicle With Underlain Device-to-Device Communications: Performance and Tradeoffs. *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS*, VOL. 15, NO. 6, JUNE 2016 3949
- [35] H. Shen, Q. Ye, W. Zhuang, W. Shi, G. Bai, G. Yang. Drone-Small-Cell-Assisted Resource Slicing for 5G Uplink Radio Access Networks Publisher: IEEE Trans on Vehicular Technology, May 2021
- [36] Q. Zhang, F. Liu, C. Zeng Adaptive Interference-Aware VNF Deployment and Migration for 5G Network Slice Publisher: IEEE Transactions on Networking, May 2021
- [37] M. Golkarifard, C. -F. Chiasserini, F. Malandrino, Ali Movaghara. Dynamic VNF placement, resource allocation and traffic routing in 5G Elsevier Computer Networks, April 2021, Vol. 188.
- [38] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications IEEE Transactions on Wireless Communications, VOL. 16, NO. 11, November 2017.
- [39] G. Faraci, A. Raciti, S. A. Rizzo, G. Schembra. Green wireless power transfer system for a drone fleet managed by reinforcement learning in smart industry. *Applied Energy* Volume 259, 1 February 2020
- [40] O. Bekkouche, M. Bagaa and T. Taleb. Toward a UTM-Based Service Orchestration for UAVs in MEC-NFV Environment. *Proc. of IEEE Globecom 2019*.
- [41] B. Nemeth, N. Molner, J. Martinperez, C. J. Bernardos, A. De la Oliva and B. Sonkoly. Delay and reliability-constrained VNF placement on mobile and volatile 5G infrastructure. *IEEE Trans. on Mobile Computing* Jan. 2021.
- [42] N. Mohamed, J. Al-Jaroodi, I. Jawhar, H. Noura, and S. Mahmoud UAVFog: A UAV-Based Fog Computing for Internet of Things. *IEEE SMART WORLD 2017*, August 2017.
- [43] M. Alharthi, A.-E. M. Taha, H. S. Hassanein Utilizing Network Function Virtualization for Drone-Based Networks *IEEE Globecom 2020*.
- [44] C. Rаметта, G. Schembra. Designing a Softwarized Network Deployed on a Fleet of Drones for Rural Zone Monitoring MDPI Sensors, March 2017.
- [45] B. Yi, X. Wang, K. Li, S. k. Das, M. Huang A comprehensive survey of Network Function Virtualization Elsevier Computer Networks, Vol. 133, 2018.
- [46] M. F. Bari, et al., Orchestrating virtualized network functions, *IEEE Trans. Netw. Serv. Manag.* PP (99) (2016) 1-14.
- [47] R. Riggio, et al., Scheduling wireless virtual networks functions, *IEEE Trans. Netw. Serv. Manag.* 13 (2) (2016) 240-252.

- [48] M.C. Luizelli, et al., Piecing together the NFV provisioning puzzle: efficient placement and chaining of virtual network functions, in: Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015.
- [49] A. Gupta, M.F. Habib, P. Chowdhury, M. Tornatore, B. Mukherjee, Joint Virtual Network Function Placement and Routing of Traffic in Operator Network, University of California Davis, 2015, pp. 1–5.
- [50] C. Pham, N.H. Tran, S. Ren, W. Saad, C.S. Hong, Traffic-aware and energy-efficient vNF placement for service chaining: joint sampling and matching approach, *IEEE Trans. Serv. Comput. PP* (99) (2017) 1–13.
- [51] B. Addis, D. Belabed, M. Bouet, S. Secci, Virtual network functions placement and routing optimization, in: Proceedings of the IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls, ON, 2015, pp. 171–177.
- [52] H. Moens, F.D. Turck, VNF-p: a model for efficient placement of virtualized network functions, in: Proceedings of the 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, 2014, pp. 418–423.
- [53] S. Khebbache, M. Hadji, D. Zeghlache, Scalable and cost-efficient algorithms for VNF chaining and placement problem, in: Proceedings of the 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), 2017, pp. 92–99.
- [54] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, C. Qiao, GREP: guaranteeing reliability with enhanced protection in NFV, in: HotMiddlebox, London, 2015.
- [55] V. Eramo, E. Miucci, M. Ammar, F.G. Lavacca, An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures, *IEEE/ACM Trans. Netw. PP* (99) (2017) 1–18.
- [56] E.J. Scheid, et al., Policy-based dynamic service chaining in network functions virtualization, in: Proceedings of the IEEE Symposium on Computers and Communication (ISCC), 2016.
- [57] G. Colajanni, D. Sciacca, “An Optimization model for service requests management in a 5G network architecture”, A. Masone et al. (eds.), *Optimization and Data Science: Trends and Applications*, AIRO Springer Series 6, 2021, 81–98.
- [58] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, “Performance Optimization for UAV-Enabled Wireless Communications under Flight Time Constraints,” *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.
- [59] M. M. Azari, G. Geraci, A. Garcia-Rodriguez and S. Pollin, “Cellular UAV-to-UAV Communications,” *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–7.
- [60] G. Faraci, C. Grasso and G. Schembra, “Reinforcement-Learning for Management of a 5G Network Slice Extension with UAVs,” *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 2019.
- [61] G. Faraci, C. Grasso, and G. Schembra. “Fog in the Clouds: UAVs to Provide Edge Computing to IoT Devices,” *2020 ACM Trans. Internet Technol.* 20, 3, Article 26 (October 2020)
- [62] D. Lee, J. Zhou and W. T. Lin, “Autonomous battery swapping system for quadcopter,” *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 118–124.
- [63] D. C. Nguyen et al., “6G Internet of Things: A Comprehensive Survey,” in *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 1 Jan.1, 2022, doi: 10.1109/JIOT.2021.3103320.
- [64] P.C. Fishburn, “Additive Utilities with Incomplete Product Set: Applications to Priorities and Assignments”. *Journal of the Operations Research Society of America*, (1967).
- [65] E. Triantaphyllou, “Multi-Criteria Decision Making: A Comparative Study”. Dordrecht, The Netherlands: Kluwer Academic Publishers. p. 320, (2000).
- [66] L.A. Zadeh, “Optimality and non-scalar-valued performance criteria”, *IEEE Trans Automat Contr AC-8*:59–60, (1963).
- [67] IBM ILOG CPLEX Optimization studio CP Optimizer user’s manual, version 12 release 8. IBM Corporation (2017).
- [68] W. Rankothge, F. Le, A. Russo, and J. Lobo, “Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms,” *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 343–356, Jun. 2017.
- [69] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, “An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.
- [70] L. Guo, J. Pang, and A. Walid, “Dynamic service function chaining in SDN-enabled networks with middleboxes,” in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.]
- [71] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li and J. Rao, “Dynamic Flow Migration for Embedded Services in SDN/NFV-Enabled 5G Core Networks,” in *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2394–2408, April 2020.