Utilizing Network Function Virtualization for Drone-based Networks

Mohannad Alharthi*‡, Abd-Elhamid M. Taha†, Hossam S. Hassanein*
*School of Computing, Queen's University, Kingston, Ontario, Canada. Emails: {harthi, hossam}@cs.queensu.ca
†Electrical Engineering Department, Alfaisal University, Riyadh, Saudi Arabia. Email: ataha@alfaisal.edu

***Abstract*—The emerging use of drones to support communication infrastructure and to deploy temporary networks in emergency scenarios is under active research. In this paper, we consider virtualizing drones' computing resources when deploying drone networks by utilizing Virtual Network Functions (VNFs) to process and deliver mission-related data traffic. This is aimed at cases where offloading and processing traffic to the cloud is not an option due to the unavailability of terrestrial and satellite communications. After discussing possible applications, we propose and evaluate a scheme for the deployment and placement of a drone network as well as the VNFs needed to deliver a set of traffics flowing from different locations in the task area.**

I. Introduction

In situations where a communication network is required for a certain task where a reliable infrastructure is not accessible, the use of drones equipped with communication capabilities to deploy a temporary network is an attractive choice [1] [2] [3]. This is especially useful for tasks in remote areas, as deploying a fixed infrastructure for temporary situations is costly. Much is gained by deploying a drone-based network due to its flexibility and relative low cost. While deploying drone-based backhaul networks for emergency situations has been discussed in existing literature, there is further merit in designing a flexible network architecture for multi-task drone network deployments. In a previous work, we discussed the design of an architecture based on Software Defined Networking (SDN) concepts that enable dynamic programmability, reuse, and cost-effective development. The capabilities of the proposed architecture extend to both communication aspects and mission-related tasks through a unified programmable interface.

In some applications such as those involving autonomous drones, the processing of complex computational tasks is often required to assist the mission. Such tasks can be offloaded to the edge and cloud infrastructure. However, some missions are deployed in remote areas that lack reliable access to edge or cloud resources while requiring processing of some tasks immediately. Given the above limitation, augmenting drones with light-weight virtualized computing capabilities made available by single-board computers can be promising for some applications. Network Function Virtualization (NFV) technology allows for the provision of softwarized network or data processing blocks known as Virtual Network Functions (VNFs) which may be easily deployed on virtualized hardware. VNFs can be used to implement various aspects such networking, operating drone sensors and flight control. NFV then can enable mission operators to reconfigure drones with multiple service using the same drone for multiple missions. This allows for promising use cases, such as deploying emergency VoIP network or video surveillance services, provided VNFs that implement such functionalities. This flexibility paves the way for promising future considerations such as dynamic adjustments of drone functions while in the air in response to changing requirements or failures.

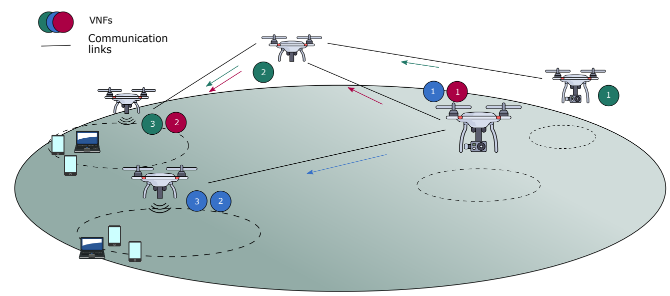


Fig. 1. An example of a drone network with sample traffics and VNFs. Circles represent individual VNFs hosted on drones, while their colors represent the individual SFCs to which they belong and numbers represent VNF order in each SFC. Arrows show the flow of traffic between VNFs in each SFC. Here, VNF traffic flows from right to left.

In order to efficiently utilize the capabilities stated above, we propose the planning of applicable network deployments by first expressing missions as a set of Service Functions Chains (SFCs) where each chain represents a traffic and a set of VNFs that process traffic and deliver it from source to target locations within the mission area, overcoming difficult terrain and large distances. Based on such requirements, optimal planning is needed to determine the size of the network, node locations, topology, and the placement of VNFs in computing resources mounted on drones. The end goal of planning is that the network is collectively capable of carrying traffic across the mission area. The contribution of this paper is a joint drone network deployment and a VNF placement scheme, with the goal of constructing a minimal and independent drone network that satisfies mission traffic requirements.

II. Related Work

VNF and SFC placement and orchestration is an actively explored area in NFV literature in cloud computing and data

‡ Mohannad Alharthi is also affiliated with King Abdulaziz University (KAU), Jeddah, Saudi Arabia

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322405

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

center setting. The survey presented in [4] provides the state-of-the-art on this area. Recent trends on this front also include [5] [6] [7].

The use of NFV in drone networks has been explored only recently. Authors in [8] describe an SDN/NFV-based architecture for UAV networks for rural zone monitoring. The Flying Ad hoc Network (FANET) provides video monitoring as a service, where cameras on the ground and on UAVs capture and stream footage of monitored rural areas. A configurable NFV-system for multi-UAV network services is described in [9]. The system enables deploying VNFs on UAVs with mounted Raspberry Pi boards forming a wireless ad-hoc network. Authors of [10] describe the use of SDN/NFV in military related missions. The NFV system uses container-based VNFs deployed at the network edge on nodes on ground. The VNFs perform mission monitoring and anomaly detection tasks. The system orchestrates the deployment and migration of VNFs that supports UAVs while they roam different areas.

III. Problem Description

In this section, we describe the use-cases that motivate this work, outline some operational concerns, provide the problem definition.

A. Use Cases

A remote monitoring mission can express its network and computation tasks as SFCs. These are composed as a series of virtual network functions that perform network and processing tasks while traffic passes through them to reach its destination (See Fig. 1). For a monitoring and video streaming mission, the VNFs can include video capture, video transcoding, and video streaming [8]. These functions can be deployed for traffic initiated from a video capturing drone, where the first VNF should be placed. The final streamer function can be placed on the same drone or a different drone depending on user location. Intermediate functions can be placed in any of those drones or intermediate ones, depending on available computing resources on drones. In this case, nodes are assumed to be mostly hovering where they are deployed. Similarly, a VoIP [9] system deployment may involve similar functions.

B. Problem Definition

The aim is to deploy a drone network with the minimum number of drones (as the nodes of the network) and to find the suitable locations for nodes to form the necessary links between them. The topology needs to be generated to support the delivery of a given set of traffic flows from their source to target locations through the links between drones. The deployment also includes the placement of the VNFs that process traffic on drones computing resources along its route from source to target while links between VNFs are within the capacity of network links. Deployment of such a network is limited by a given budget in terms of the number of available drones to deploy and their computing capacities.

Considerations: NFV-based networks typically assume and benefit from dynamic orchestration and dynamic instantiation migration of functions. However, in this paper we only consider the initial or static deployment with an already known set of function requirements and traffics. Furthermore, we assume in this architecture that there are sufficient computing resources available on-board the drones. For instance, single-board computers such as the Raspberry Pi can be used along with container-based virtualization [11] to host VNFs, as it is a light-weight alternative to full virtual machines. The proposed scheme in this paper is assumed to be implemented by a planning and initialization module. VNF images are preloaded and configured on the on-board computers and instantiated at deployment time.

IV. System Model

The aim is to construct the physical topology for the network to be deployed from the set of available drones $D = \{d_1, d_2, \dots\}$ and all possible directed links between them $E = \{(d_i, d_j) \mid d_i, d_j \in D\}$. The topology of the constructed network is represented by a directed graph $G = (\bar{D}, \bar{E})$, where $\bar{D} \subseteq D$ is the set of selected drones (nodes) and $\bar{E} \subseteq E$ is the set of selected links. The set of all allowable drone locations is denoted by $L = \{l_1, l_2, \dots\}$. All locations are in a 2-D horizontal plane assuming a predetermined altitude.

The set of all SFCs (hereafter called *traffics*) is denoted by $R = \{1, 2, \dots\}$. Each $r \in R$ is associated with a directed linear graph $S_r = (F_r, VL_r)$, which represents the chain's ordered VNFs $F_r = \{f_1, f_2, \dots f_{|F_r|}\}$ and the VNF links $VL_r = \{(f_m, f_{m+1}) \mid m = 1, \dots, |F_r| - 1\}$ that connect them. We denote by $s_r, \tau_r \in L$ the source and target locations of traffic r. The required link throughput for any traffic r is denoted by δ_r (in Mbps), while cpu_{f_m} and ram_{f_m} represent the CPU cores and RAM capacities required by VNF f_m.

Each $d_i \in D$ has a known capacity of compute resources in terms of the available CPU cores and RAM, expressed as cpu_{d_i} and ram_{d_i}, respectively.

We adopt the free space propagation model to model wireless links between drones due to the lack of obstacles in drone altitudes in remote areas. The path loss, in dB, over transmission distance $dist$ in meters is defined as [12]:

$$PL_{dist} = FSPL_{dist} - \eta_{LOS}$$

where $FSPL_{dist}$ is the free space path loss defined as:

$$FSPL_{dist} = 20\log_{10}(dist) + 20\log_{10}(f_c) - 147.55$$

where f_c is the carrier frequency in Hz and η_{LOS} is the additional line-of-sight attenuation due to the environment. Thus, for any given transmission power, denoted by P_t, the power of the received signal in dBm at the receiver is $P_r = P_t - Pl_{dist}$.

The capacity of the wireless channel in bits per second is given by the Shannon capacity equation [13]: $C = Blog_2(1 + \frac{P_r}{P_n})$, where B is the channel bandwidth in Hz, P_r and P_n are the powers of the received signal and noise in watts respectively.

Based on the channel model above, we define the following constants for every pair of locations in L. We define $pl_{l_k,l_l} \in$

$\{0,1\}$ as equal to 1 if the path loss between a pair of nodes located at $l_k, l_l \in L$ is below the path loss threshold pl_{max}. We also define β_{l_k,l_l} as the achievable link capacity between the pair of nodes placed at l_k, l_l.

A. Deployment Problem Formulation

We formulate the deployment problem as a linear program that constructs the graph G, based on a given R. We are also given the set of nodes (drones) D and their capacities, which represent the budget of available drones.

The decision variables determine the location of drones and the links formed between them (physical links) as well as the placement of VNFs on each drone and the assignment of VNF links to physical links. The decision variables are:

- $u_{d_i,l_k} = \{0,1\}$ indicates if d_i is deployed to l_k.
- $u_{d_i} = \{0,1\}$ is derived from the previous variable and it indicates whether drone d_i is deployed regardless of selected location.
- $k_{d_i,d_j} = \{0,1\}$ indicates if the physical link (d_i, d_j) is formed between the pair of drones.
- $x_{f_m,d_i} = \{0,1\}$ indicates if VNF f_m is placed in d_i.
- $y^{f_m,f_{m+1}}_{d_i,d_j} = \{0,1\}$ indicates if VNF link (f_m, f_{m+1}) is using the physical link (d_i, d_j).

Our goal is to minimize the number of deployed drones to reduce the mission cost, and have the shortest possible paths for traffics. Thus the objective function is:

$$\min \sum_{d_i \in D} u_{d_i} + \sum_{r \in R} \sum_{\substack{(f_m, f_{m+1}) \\ \in VL_r}} \sum_{\substack{(d_i,d_j) \\ \in E}} y^{f_m f_{m+1}}_{d_i d_j} \tag{1}$$

subject to

$$\sum_{l_k \in L} u_{d_i,l_k} = u_{d_i}, \quad \forall d_i \in D \tag{2}$$

$$\sum_{d_i \in D} u_{d_i,l_k} \leq 1, \quad \forall l_k \in L \tag{3}$$

$$z(d_i, l_k, d_j, l_l) = 1 \quad \text{iff} \quad u_{d_i,l_k} + u_{d_j,l_l} + pl_{l_k,l_l} = 3 \qquad \forall (d_i, d_j) \in E, \quad \forall (l_k, l_l) \in L \tag{4}$$

$$k_{d_i,d_j} \leq \sum_{(l_k,l_l) \in L} z(d_i, l_k, d_j, l_l), \quad \forall (d_i, d_j) \in E \tag{5}$$

$$x_{f_m,d_i} \leq u_{d_i}, \quad \forall r \in R, \forall f_m \in F_r, \forall d_i \in D \tag{6}$$

$$\sum_{d_i \in D} x_{f_m,d_i} = 1, \quad \forall f_m \in F_r, \forall r \in R \tag{7}$$

$$\sum_{r \in R} \sum_{f_m \in F_r} x_{f_m,d_i} \times cpu_{f_m} \leq cpu_{d_i}, \quad \forall d_i \in D \tag{8}$$

$$\sum_{r \in R} \sum_{f_m \in F_r} x_{f_m,d_i} \times ram_{f_m} \leq ram_{d_i}, \quad \forall d_i \in D \tag{9}$$

$$y^{f_m,f_{m+1}}_{d_i,d_j} \leq k_{d_i,d_j} \qquad \forall r \in R, \forall (f_m, f_{m+1}) \in VL_r, \forall (d_i, d_j) \in E \tag{10}$$

$$\sum_{r \in R} \sum_{f_m \in F_r} y^{f_m,f_{m+1}}_{d_i,d_j} \times \delta_r \leq \sum_{(l_k,l_l) \in L} z(d_i, l_k, d_j, l_l) \times \beta_{l_k,l_l} \qquad \forall (d_i, d_j) \in E \tag{11}$$

$$\sum_{(d_i,d_j) \in E} k_{d_i,d_j} \leq \gamma_{max}, \quad \forall d_i \in D \tag{12}$$

$$\begin{aligned} &\text{if} \quad x_{f_1,d_i} = 1 \quad \text{then} \quad u_{d_i,s_r} = 1 \\ &\text{if} \quad x_{f_{|F_r|},d_i} = 1 \quad \text{then} \quad u_{d_i,\tau_r} = 1 \\ &\qquad \forall r \in R, \forall d_i \in D \end{aligned} \tag{13}$$

$$\sum_{d_j \in n(d_i)} (y^{f_m,f_{m+1}}_{d_i,d_j} - y^{f_m,f_{m+1}}_{d_j,d_i}) = x_{f_m,d_i} - x_{f_{m+1},d_i} \qquad \forall r \in R, \forall (f_m, f_{m+1}) \in VL_r, \forall d_i \in D \tag{14}$$

In the above formulation, constraints (2) and (3) ensure that a drone is assigned to one location, and each location is assigned to one drone only.

In (4), $z(d_i, l_k, d_j, l_l)$ is a decision variable used as a flag equal to 1 only if a pair of drones d_i, d_j, can achieve the required path loss pl_{max} in their assigned locations l_k, l_l (in any combination). In constraint (5), a link (d_i, d_i) is selected based on the flag values from (4) for all locations of (d_i, d_j).

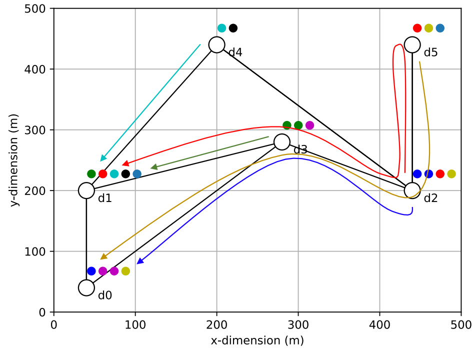
The constraints for placing VNFs are as follows. Constraints (6) and (7) ensure that a VNF is placed on a deployed drone, and that each VNF is placed on one drone only. In (8) and (9), we ensure that placed VNFs obey the CPU and RAM capacities of the drone on which they are placed. Constraint (10) ensures that VNF links are placed on selected links, while constraint (11) ensures that VNF links are within the capacity of the link they are assigned. Constraint (12) limits the number of links a single drone can form to γ_{max}, according to drone capabilities.

Constraint (13) ensures that VNF placement location requirements match the location of the drone hosting the VNF. The constraint (14) is the flow conservation constraint which balances the traffic flow and ensures that VNF link placement is according to the placement of corresponding VNFs [14]. We denote by $n(d_i)$ nodes adjacent to d_i in E.

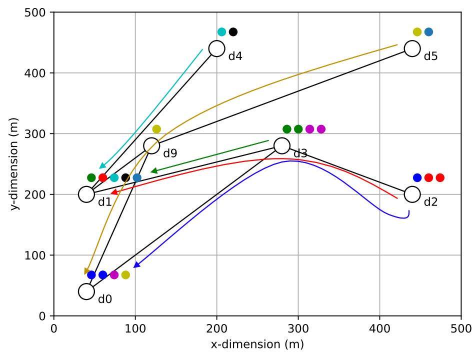
V. Performance Evaluation

A. Evaluation Setup

In this section, we present the performance evaluation of our proposed scheme and results. The simulation and deployment scheme were implemented in Python and CPLEX (version12.9). We assume 10 drones are available for deployment, with fixed CPU and RAM capacities as shown in Table I. The channel model parameters used by the deployment scheme are also shown in Table I. The deployment area size is $500m \times 500m$. For tractability, the placement locations in L are discretized into a grid with cells each of size $w \times h$. The size of cells can be varied according to the density required for



(a) Minimal drones (baseline)



(b) Minimal drones and VNF paths

Fig. 2. Topology and placement results sample for 8 traffics. Circle shapes represent drones, while filled circles represent VNFs placed on drones. VNFs of the same color belong to the same traffic. The two leftmost drones are placed in target locations for traffics, while other nodes are sources or relays. Arrows represent the path allocated for select traffics in corresponding color.

the task at hand and the size of the area. Here it is $80m \times 80m$. For each pair $l_k, l_l \in L$, we compute β_{l_k,l_l} according the propagation and capacity formulae described in IV.

We evaluate the proposed scheme in terms of the size of the produced network and its utilization while varying the number of traffics $|R|$ from 2 to 8. For each evaluation of $|R|$, traffics, their locations and throughput requirements as well as the configurations of associated VNFs are drawn randomly from the ranges shown in Table I. In each evaluation instance, $|R|/2$ locations for each traffic are selected from L as a pool of source locations $L_{source} \in L$, and two locations from one side of the area as a pool for target positions L_{target}. For each generated traffic r, we draw randomly from L_{source} and L_{target} and assign these to s_r and τ_r $\forall r \in R$ respectively. As a result, some source and target locations are shared among multiple traffics. Results are compared with a baseline scheme to show the effects of the objective function. The baseline scheme drops the second term of the objective function, i.e., it only minimizes the number of drones. Runs for each evaluation are repeated 10 times, and the averages of results are recorded.

B. Results

An example of a network topology and placement of 8 traffics is shown in Fig. 2. Fig. 2a shows a network produced by the baseline scheme and Fig. 2b shows another network

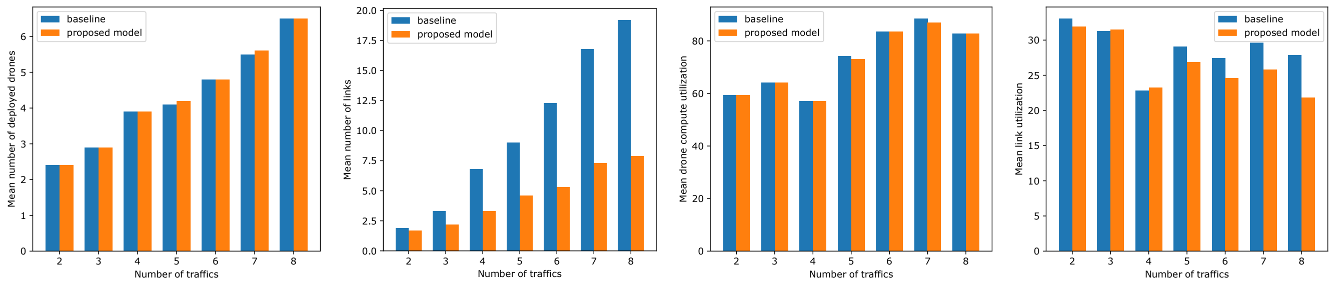
TABLE I
EVALUATION PARAMETERS

Parameter	Value
Network Parameters	
Transmission power P_t	25 dBm
Carrier frequency f_c	2 GHz
Channel bandwidth B	20 MHz
LOS attenuation η_{LOS}	5 dB
pl_{max}	80 dB
Noise power P_n	-60 dBm
Number of drones $\|D\|$	10 drones
Deployment Area	
Area	500m $\times$ 500m
L cell size $w \times h$	80m $\times$ 80m
Drone Capacities	
Drone CPU cores CPU_{d_i}	8 cores
Drone RAM RAM_{d_i}	16 GB
Max. number of links per drone γ_{max}	3
Traffic Parameters	
Number of traffics $\|R\|$	$[2, 8]$
CPU_{f_m}, RAM_{f_m}	$[1, 3]$ cores, $[1, 3]$ GB
VNFs per traffic F_r	$[2, 4]$
Traffic throughput δ_r	10, 20, 25, 40 Mbps

produced by the proposed scheme. In both instances, the two leftmost drones are deployed to target locations, meaning that the last functions of all traffics are placed in these two drones, while remaining nodes are sources or relays of traffics. It can be observed that some functions are placed along the route of traffics while other functions are placed on a single node. Drone computing resources are shared between functions of all traffics, and drones are positioned to support the delivery of traffics from source to target locations. The baseline results in more hops for traffics and may introduce routing cycles which makes it inadequate for practical use. The proposed scheme avoids cycles without requiring additional constraints.

Fig. 3 shows the properties of the produced networks and their trends against the number of traffics. Fig. 3a shows that networks increase in size with the number of traffics, as each traffic has its own set of VNFs that require compute and network resources. Satisfying location requirements for traffics also impacts the network size, as each traffic has location requirements that may necessitate the deployment of additional drones. The proposed scheme is equal to the baseline in this regard as both minimize the number of drones. Fig. 3b illustrates that the number of links also increases with the number of traffics. However, optimizing for minimum paths for traffics requires considerably fewer physical links than the baseline. Note that we count the number of utilized directed links. Fig 3c shows the utilization of drone computing resources. Only the CPU utilization is reported, which is the percentage of the CPU cores used by VNFs per drone. Fig. 3d shows the link utilization. For low traffics, the proposed and baseline schemes show the same link utilization. However, the utilization tends to fall below that of the baseline as the traffics increase due to utilizing shortest paths. Note that the proposed scheme produces a lower number of links than the baseline as shown in Fig 3b.

Fig. 4 shows the VNF compute distribution (top) and the average number of hops for traffics (bottom). The top graph



(a) Deployed drones (b) Number of links (c) Drone utilization (d) Link utilization

Fig. 3. Results of drone deployment and VNF placements while varying the number of traffics between 2 and 8 traffics using the baseline scheme (minimal number of drones) and the proposed scheme (minimal drones and traffic paths for VNF traffics)

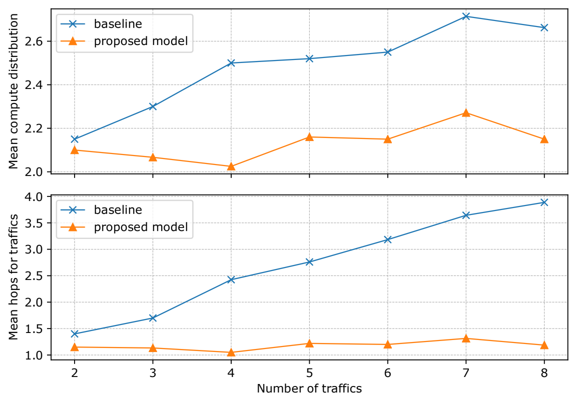


Fig. 4. Average distribution of traffics' VNFs over nodes (top) and average hop count per traffic (bottom)

shows the spread of VNFs of a traffic over multiple drones, i.e., the number of drones the VNFs of a traffic are placed in. The bottom graph shows the average number of hops for traffics from the first VNF (in the source location) to the last one (in a target location). These measures do not necessarily correlate, because a traffic with 4 VNFs can have its VNFs split between two nodes, but still travels 3 hops from source to target, with some nodes acting as relays for this particular traffic. All traffic VNFs are distributed among two or three drones as shown in the top graph of Fig. 4. Traffics placed according to the proposed scheme results in single hops. Single hop links are easily formed given the small deployment area; however, in a larger area the number of hops should be minimized.

VI. Conclusion

In this paper, we presented a joint deployment and placement scheme for drone networks with VNFs hosted on drones computing resources. The deployment determines the minimal network topology based on supplied traffics and associated VNFs along with location requirements. The formed topology supports the delivery of traffics and the hosting of the VNFs on its own computing resources. We demonstrated the feasibility of the idea of deploying a drone network in terms of functions required to process traffic with location requirements. Our scheme is applicable to drone missions in challenging scenarios, in areas lacking access to communications infrastructure required to offload processing functions.

Acknowledgment

This work was generously supported in part by Alfaisal University through its Office of Research and Graduate Studies.

References

[1] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[2] V. M. R. E. A. Estepa and G. Madinabeitia, "Deploying a reliable uav-aided communication service in disaster areas," *Wireless Communications and Mobile Computing*, 2019.

[3] M. Y. Selim and A. E. Kamal, "Post-disaster 4g/5g network rehabilitation using drones: Solving battery and backhaul issues," in *IEEE Globecom Workshops*. IEEE, 2018, pp. 1–6.

[4] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Trans. Netw. Service Manag*, vol. 13, no. 3, pp. 518–532, 2016.

[5] M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, "Joint energy efficient and qos-aware path allocation and vnf placement for service function chaining," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 374–388, March 2019.

[6] B. Spinnewyn, J. F. Botero, C. Donato, and S. Latré, "Effective nfv orchestration for wide-ranging services across heterogeneous cloud networks," in *IFIP/IEEE Symp. on Integrated Network and Service Management (IM)*, April 2019, pp. 107–115.

[7] A. Mohamad and H. S. Hassanein, "On demonstrating the gain of sfc placement with vnf sharing at the edge," in *IEEE Global Commun. Conf. (GLOBECOM)*, May 2019.

[8] C. Rametta and G. Schembra, "Designing a softwarized network deployed on a fleet of drones for rural zone monitoring," *Future Internet*, vol. 9, no. 1, p. 8, 2017.

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

[10] K. J. S. White, E. Denney, M. D. Knudson, A. K. Mamerides, and D. P. Pezaros, "A programmable sdn+nfv-based architecture for uav telemetry monitoring," in *14th IEEE Ann. Consumer Commun. Netw. Conf. (CCNC)*, Jan 2017, pp. 522–527.

[11] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and linux containers," in *IEEE Int. Symp. Performance Analysis Systems and Software (ISPASS)*, March 2015, pp. 171–172.

[12] U. Challita and W. Saad, "Network formation in the sky: Unmanned aerial vehicles for multi-hop wireless backhauling," in *IEEE Global Commun. Conf. (GLOBECOM)*. IEEE, 2017.

[13] A. Goldsmith, *Wireless Communications*. Stanford University, 2005.

[14] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 725–739, 2016.