

Quality Aware Aerial-to-Ground 5G Cells through Open-Source Software

Francesco D'Alterio[†], Ludovico Ferranti^{*†}, Leonardo Bonati^{*}, Francesca Cuomo[†], Tommaso Melodia^{*}

^{*}Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02115, USA

[†]Sapienza University of Rome, 00185 Rome, Italy

Email: {ferranti.l, bonati.l, melodia}@northeastern.edu

Email: francesca.cuomo@uniroma1.it - dalterio.1771576@studenti.uniroma1.it

Abstract—This paper investigates the advantages and design challenges of leveraging Unmanned Aerial Vehicles (UAVs) to deploy 4G/5G femto- and pico-cells to provide quality-aware user service and improve network performance. In order to do so, we combine UAVs dashing flight capabilities with Software-defined Radios (SDRs) flexibility and devise the concept of self-optimizing UAV Base Stations (UABSs). The proposed framework allows for on-the-fly drone repositioning based on rigorous optimization techniques using real-time network metrics to enhance users' service. This makes it possible to offload the traditional cellular infrastructure, or to mend its temporary failure, by deploying UABSs in areas of interest. Cellular connectivity is, then, provided to mobile subscribers through the LTE-compliant OpenAirInterface software interfaced with the on-drone SDR.

We first describe the UABS design challenges and approaches. Then, we give details on the devised optimization algorithm and its main requirements. Finally, we illustrate a prototype implementation of the proposed UABS that leverages an SDR device and a PX4 flight controller, and test its effectiveness. Experimental results demonstrate that UABSs are able to autonomously reposition themselves based on cellular network metrics and to improve network performance.

Index Terms—Software-defined Networking, 5G, UAV Networks, Optimization.

I. INTRODUCTION

The last few years have been seeing Unmanned Aerial Vehicles (UAVs) opening new branches of research. Their inherent flexibility and versatility, in fact, make it possible to easily deploy them on-demand to improve existing infrastructure capabilities or to mend its temporary failure. Specifically, the combination of UAVs and Software-defined Radios (SDRs) allows the creation of flying cellular Base Stations (BSs) able to enhance network connectivity. In this way, pico- and femto-cells can be deployed on-demand to improve the service supplied by inflexible cellular macro-cells (e.g., offloading them) or to provide connectivity to users in remote areas.

The vast majority of the existing work on drone control solutions focuses on autonomous drone deployments based on external sources of information, such as GPS positioning systems or 3D maps of the surrounding area. However, these solutions are tightly bound to the availability of these external resources (e.g., GPS signals), and often increase system overhead, latency and cost. Other approaches, relying on UAV measurements and sensor readings, thus, may be preferable in time-critical applications or resource-constrained environ-

ments (e.g., in which external sources of information are not readily available).

In this paper, we propose a quality-aware aerial-to-ground cellular framework involving a UAV operating as standalone base station and core network. These Unmanned Aerial Base Stations (UABSs) are used to extend an existing terrestrial infrastructure or to compensate for temporary cell overload or outage. The developed model, relying on real-time Quality of Service (QoS) measurements, either directly derived by the UABS or sent by mobile users as feedback, provides a fast gradient-based algorithm for online drone position optimization. Unlike most of the existing solutions, our approach aims at jump-starting the UABS deployment and positioning without relying on any source of information external to the UABS-User Equipments (UEs) environment.

We prototyped the UABS, using an off-the-shelf quadrotor UAV equipped with an open-source PX4 flight controller and general purpose compute board connected to an SDR. The compute board runs OpenAirInterface, implementing standard-compliant LTE and 5G protocol stacks, and jointly serves as cellular base station and core network.

Several challenges arise while devising such system, among which: (i) Improving the UAV capabilities with a standard-compliant cellular communication system; (ii) designing optimization algorithms tailored to resource-limited UAV compute boards and that rely on a limited amount of information based on BS-UEs parameters only; (iii) dealing with the UAV limited power capabilities, and (iv) carefully choosing hardware equipment not exceeding the drone weight payload. While addressing these challenges, we offer the following contributions: We (i) design, prototype and test a 5G-compliant Unmanned Aerial Base Station; (ii) provide an open-source framework for autonomous positioning of UABSs, and (iii) propose an algorithm for real-time UABS repositioning based on network parameters able to effectively optimize network performance.

The rest of this paper is organized as follows. In Section II, we discuss the related literature work regarding the use of UAVs in cellular networks. In Section III, we illustrate the UABS system design, while in Section IV we present our framework. In Section V and Section VI, we describe our UABS prototype, and experimentally evaluate its performance. Finally, Section VII briefly discusses some key points emerged during this work, while Section VIII concludes our work.

II. RELATED WORK

The use of UAVs to provide and strengthen mobile connectivity opens a rather unexplored branch of research. Zeng et al. provide an overview of the features, and introduce key design aspects of UAV-based wireless communication systems in [1]. Works considering coverage analysis in specific environments include those of Nguyen et al. [2], and Greenberg and Levy [3] with rural and urban areas, respectively.

Multiple approaches exist for drone flight and positioning in aerial networks. De Silva and Rajasinghege precompute distances between a UAV and points of interests to reduce the flying path in ad-hoc UAV networks [4], while Cheng et al. optimize UAV trajectories whilst offloading traffic among adjacent BSs [5]. Wang et al. present a 2-steps UAV positioning strategy which minimizes the number of required UAVs and their position using centralized and distributed approaches, respectively [6]. Kalantari et al. consider multiple centralized approaches based on the backhaul link throughput to optimize the position of a drone-BS in [7], while Dixon and Frew describe a least-squares gradient-based optimization algorithm for UAV relays in 802.11b/g networks in [8]. Chakareski et al. provide a theoretical framework to efficiently find coverage radius, and perform energy efficient radio resource management for UAV-assisted mmWave cellular networks [9]. Gangula et al., instead, deal with an optimization framework for a UAV-mounted LTE relay [10]. Similar to our approach, the UAV position is computed maximizing the downlink throughput of mobile subscribers but using users GPS locations and a 3D map of the surrounding environment, instead.

Other works leverage UAVs to deploy, establish, characterize, and optimize network connectivity. For instance, Ladosz et al. aim at maintaining network connectivity through UAV relays [11], while Zhan et al. propose an algorithm to optimize link quality between ground stations and relays by controlling the UAV heading angle [12]. Gapeyenko et al. mathematically investigate mmWave spectrum phenomena during 5G backhaul operations for UAV-based relays [13]. Bertizzolo et al., instead, experimentally evaluate the benefits of location-aided mmWave backhaul link management for UAV-based mobile cells [14]. Lyu et al. minimize the number of UAVs needed to cover an area of interest [15], while Iellamo et al. introduce clustering algorithms to maximize UAVs capacity boost [16].

Finally, video-streaming QoS-based drone positioning is investigated in [17], while UAVs for disaster scenarios are evaluated theoretically by Deruyck et al. in [18], and experimentally by Ferranti et al. in [19].

III. SYSTEM DESIGN

While the majority of cellular network implementations remains proprietary, providing an open-source UABS platform represents the main focus of this work. To this purpose, the OpenAirInterface (OAI) software, that interfaces with a lightweight SDR mounted on the drone, is leveraged for a standard-compliant cellular base station, while the PX4 flight controller enables smooth UAV operations.

A. Mobile connectivity

We adopt OpenAirInterface in order to provide cellular connectivity to ground users [20]. This is an open-source experimental software suite providing standard compliant implementations of 4G and 5G networks. Specifically, it includes a Radio Access Network (OAI-RAN) with LTE and 5G NR base stations, as well as a Core Network (OAI-CN) implementation. The Core Network features services such as Mobile Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (SWG), and Packet Data Network Gateway (PGW). UEs are equipped with programmable SIM cards subscribed to OAI-CN that allow them to swiftly attach to the UABS, which provides them IP connectivity.

Fig. 1 shows a high-level representation of the proposed architecture. It is worth noting that the UABS embeds all the main functionalities of 4G/5G protocol stacks, thus making our solution compliant to the 3GPP guidelines and compatible with the existing cellular infrastructure.

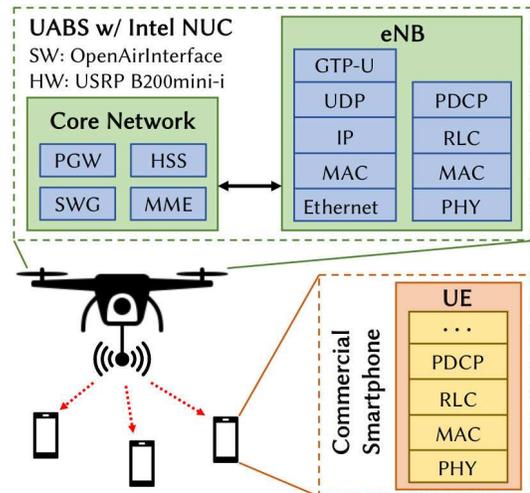


Fig. 1: High-level UE and OAI-based UABS protocol stacks.

B. Flight control

The main component governing the UAV movement is the Flight Controller Unit (FCU) that allows to operate the drone either through a remote controller or ground station commands. Moreover, it enables autonomous flight capabilities, such as the one considered in this work, by supporting advanced flight algorithms. We use the PX4 Flight controller [21] due to its high modularity and extensibility both in terms of hardware and software. The communication protocol employed by the FCU, instead, is the MAVLink messaging protocol [22]. This follows a hybrid publish-subscribe and point-to-point scheme in which data streams are sent (published) as topics relative to distinct sub-protocols (e.g., mission protocol, parameter protocol, etc.) specific for different aspects of the drone management.

IV. AUTONOMOUS POSITIONING FRAMEWORK

In this section, we describe the designed optimization framework together with the main functional blocks to develop and extend aerial mobile Heterogeneous Networks (HetNets).

Let us recall some of our system requirements: (i) All the computations need to be performed by the UABS on-board computer; (ii) the UABS optimization problem needs to be solved only using parameters and directives belonging to the UABS-UE system (i.e., without using any external source of information), and (iii) the framework deployment overhead needs to be negligible. The first aspect challenges the overall framework to be executed within a memory-, computational- and power-constrained system, such as that typical of commercial UAVs. The second and third ones, instead, force us to fine-tune the UABS positioning without using any third-party function. This also allows us to maintain full control of our system.

A. Placement Algorithm

The placement algorithm used in our framework is a variant of the Simultaneous Perturbation Stochastic Approximation method (SPSA) [23]. This is a greedy gradient-based algorithm that does not require full-knowledge of the optimization function gradient. In fact, this algorithm manages to approximate the underlying gradient by using only few objective function measurements regardless of the problem size. This feature and the algorithm's ability to deal with noise-corrupted environments make it particularly suitable for UAV applications. Specifically, during each algorithm iteration the next position of the UAV is computed by using the current one and its gradient function value as follows:

$$\theta_{k+1} = \theta_k + a_k g(\theta_k), \quad (1)$$

where k is the current algorithm iteration and θ_k represents the UAV position. The value the gradient of the objective function assumes in θ_k is given by $g(\theta_k)$, while a_k is a proportionality constant. The gradient estimation is computed as

$$\hat{g}_k(\theta_k)_j = \frac{L(\theta_k + c_k \Delta_k) - L(\theta_k - c_k \Delta_k)}{2c_k (\Delta_k)_j} \quad \forall j \in [1, p], \quad (2)$$

where L is the measured objective function, j is the considered dimension, while p is the number of involved spatial dimensions. The product $c_k \Delta_k$ denotes the displacement from the current position, where c_k is a small positive number decreasing with the number of iterations, while $\Delta_k = (\Delta_{k_1}, \Delta_{k_2}, \dots, \Delta_{k_p})^T$ is the perturbation vector.

It is worth noting that this algorithm can estimate function gradients by taking in input few measurements performed during each algorithm iteration only, regardless of the objective function size. This is particularly useful when dealing with noise-affected and instantaneous network parameters for which a corresponding gradient function cannot be computed in polynomial time. Furthermore, being our algorithm able to effectively guarantee a good trade-off between convergence time and required measurements, it is particularly suitable for UAV scenarios. In fact, although being a greedy approach, it is more likely to converge to the global optimum than classic gradient descent-based algorithms. This is due to the involved noisy measurements that allow it to avoid local optima and

saddle points with high probability [24]. Also, it requires low memory and computational capabilities, making it particularly appealing for resource-constrained drone compute boards.

During each algorithm iteration, gradients relative to pair of points close to each other are computed. Due to the short distance between them, a sudden change of the estimated measure is unlikely to happen, allowing us to consider slowly time-changing objective functions. Finally, cases in which the algorithm reliability may decrease are prevented by carefully tuning the algorithm parameters. If not handled correctly, in fact, such cases may lead to significant differences in the magnitudes of subsequent points, thus undermining the optimization algorithm results.

B. Objective Function

One of the most critical parts of the developed framework is the objective function design. This needs to take into account information on the UABS-UEs interactions and to be solvable by the chosen optimization algorithm. In order to select proper functions and parameters, extensive experiments concerning downlink traffic generated by the UABS toward the connected UEs have been performed. The considered metrics involved jitter, packet error rate, throughput, power headroom, downlink data and uplink control channel SNR, and channel quality indicator relative to each UABS-UE pair. Among these, downlink throughput proved to be the most meaningful metric for our optimization problem since it allows the UABS to be aware of how well it is serving mobile subscribers.

Even though the devised framework is general and allows us to take into account additional metrics (i.e., QoS, MAC-layer user scheduling parameters, path loss, etc.), this would come at the cost of a higher computational complexity. However, computational boards lightweight enough to be embedded in commercial UAVs are performance- and energy-constrained, thus demanding a low optimization computational complexity. For illustrational purposes, thus, in this work we only consider the UABS-UEs downlink throughput, leaving the integration of additional metrics to future work and more mature drone boards. Although by taking into account this metric only, we successfully manage to demonstrate the improved network performance achieved by our system.

The objective function we considered is as follows:

$$L(\theta_k) = \min\{M_1(\theta_k), M_2(\theta_k), \dots, M_r(\theta_k)\}, \quad (3)$$

where $M_i(\theta_k)$ represents the value of the chosen metric measured by the i -th UE, while r is the total number of UEs involved in the optimization procedures. This function is leveraged in Eq. 2 to estimate the gradient in the current position θ_k , which is, then, used in Eq. 1 to compute the next UAV position. We would like to remark the presence of the positive sign in Eq. 1, which allows the algorithm to move toward a point that maximizes the underlying optimization function. This favors the UE with the smallest throughput and guarantees user *fairness*.

C. System Parameters

Our optimization algorithm requires the characterization of the parameters a_k , c_k and Δ_k of Eqs. 1 and 2 while considering the objective function convergence properties [25]. By following Spall's SPSA directives [26], we define a_k and c_k as

$$a_k = \frac{\hat{a}}{(A + k + 1)^\alpha}, \quad (4)$$

$$c_k = \frac{\hat{c}}{(k + 1)^\gamma}, \quad (5)$$

where $A < 0.1K^{\max}$, with K^{\max} being the maximum number of algorithm iterations. We choose $K^{\max} = 1000$ and $A = 99$. The values for α and γ , instead, are $\alpha = 0.602$ and $\gamma = 0.101$ if $k \leq k_{\text{sw}}$, and $\alpha = 1$ and $\gamma = \frac{1}{6}$ otherwise, where k is the current iteration, while $k_{\text{sw}} = 0.1K^{\max}$. In our experiments, we set $k_{\text{sw}} = 100$. This is done in order to speed up the convergence procedures during the first k_{sw} iterations, as suggested in [23], and then stabilize α and γ to their optimal values [25, 27]. Values for \hat{a} and \hat{c} , instead, are empirically derived and adapted during different trials in order to obtain a good trade-off between convergence speed and number of iterations to reach convergence. Finally, suitable candidates for Δ_k must have all the Δ_{k_j} components independent and symmetrically distributed around zero, as well as finite inverse first and second moments [23]. In this case, a Rademacher distribution was adopted.

V. AERIAL BASE STATION PROTOTYPE

In this section, we describe our UABS prototype along with its three main components: The Control Station, the Computing Board and the Pixhawk FCU. The UABS prototype diagram is shown in Fig. 2.

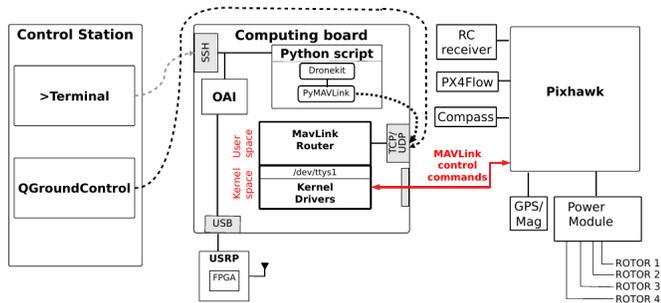


Fig. 2: UABS prototype diagram.

The Control Station is the center of operations from which new commands and directives can be sent to the UABS (e.g., new missions, utility functions, additional constraints, etc.). It runs the QGroundControl mission planning tool and it is wirelessly connected to the UABS Computing Board.

The Computing Board is the very core of the UABS. It features a MavLink Router able to interact with the Control Station QGroundControl tool via TCP/UDP connections. This component receives flying commands to be, then, interchanged

with the Pixhawk. Furthermore, the Computing Board leverages the OAI software to run an all-in-one LTE-compliant eNB and Core Network implementation, and provides IP connectivity to ground users attached to it. Additionally, it embeds a Python framework that sends UDP data streams to the connected UEs, receives their feedback measurements, and reads OAI statistics. All these measurements are, then, fed to the optimization algorithm and taken into account to compute the optimal drone location. This framework also marshals packets between the optimization program, the MAVLink Router, and the board kernel drivers by using reconfigurable APIs such as PyMAVLink and Dronekit. Finally, a software-defined radio enables communication among OAI and mobile subscribers.

The PX4 Pixhawk FCU gets MAVLink control commands from the Computing Board MAVLink Router and interacts with the drone itself. This component reads data from the drone sensors (e.g., compass, GPS, magnetometer, etc.) and regulates rotors thrust and spinning velocity to adjust the UABS position according to the found optimization results.

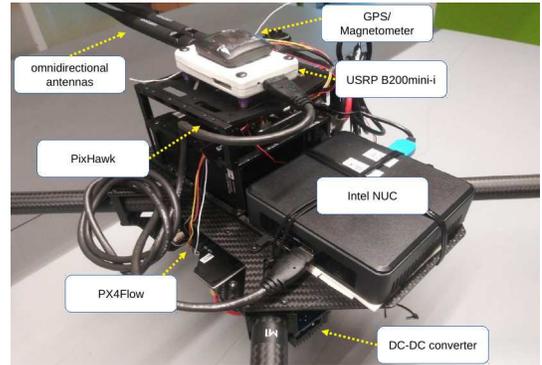


Fig. 3: UABS prototype implementation.

Our UABS prototype is shown in Fig. 3. A DJI Matrice 100 quadcopter is used as drone frame, on top of which an Intel NUC running Ubuntu Linux 16.04 LTS operates as Computing Board. A PX4 Pixhawk FCU is mounted on the drone and communicates with on-board sensors, such as a PX4Flow optical flow smart camera, allowing for a fine-tuned indoor drone flight. As for the employed SDR, the UABS embeds an Ettus Research Universal Software Radio Peripheral (USRPG) B200mini-i running the USRP Hardware Driver (UHD) FPGA software. The USRP is equipped with VERT2450 omnidirectional antennas and enables LTE-compliant communication between the UABS and its subscribers through the OAI software that runs on the Intel NUC board. Finally, additional sensors such as GPS and magnetometer are placed on top of the USRP and connected to the FCU.

As UEs, we use commercial Android smartphones, such as Samsung Galaxy S6 edge and Wiko Fever 4G. These are equipped with programmable SIM cards subscribed to the OAI Core Network, and allow them to promptly attach to the OAI eNB. Finally, a Linux workstation running Ubuntu 18.04 LTS is used as ground Control Station.

VI. EXPERIMENTAL RESULTS

We performed several experiments in order to test the effectiveness of our system. Values of the used parameters are given in Table I.

TABLE I: Experiment Parameters.

| Parameter | Value |
|------------|--|
| a_k | $\frac{\hat{a}}{(A+k+1)^\alpha}$ |
| c_k | $\frac{\hat{c}}{(k+1)^\gamma}$ |
| \hat{c} | 2 |
| \hat{a} | 10^{-3} |
| α | 0.602 if $k \leq k_{sw}$, 1 otherwise |
| γ | 0.101 if $k \leq k_{sw}$, $\frac{1}{6}$ otherwise |
| A | 99 |
| k_{sw} | 100 |
| K^{\max} | 1000 |

We considered a scenario with one UABS serving downlink data traffic to one connected UE. The UABS improves the provided service optimizing its position according to the measured metrics discussed in Section III. Since LTE transmissions are involved, experiments have been performed in a $20 \times 20 \times 4$ m anechoic chamber, shown in Fig. 4a.

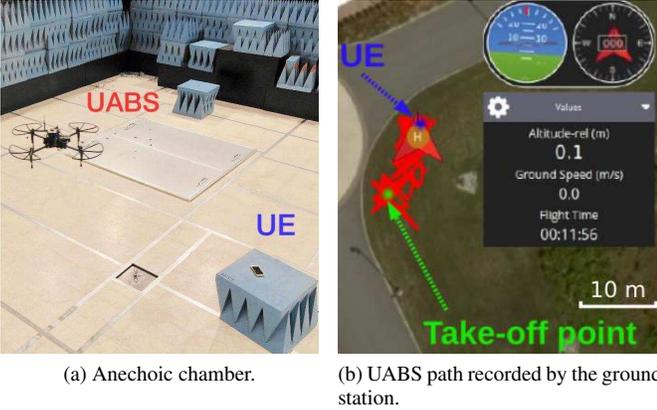


Fig. 4: UABS experimental setup.

The UABS periodically sends UDP downlink data to the connected UE through the `iperf3` Linux tool at the constant bit rate of 10 Mbps. The user, then, measures statistics of the received data and communicates them as a feedback to the UABS. The height of the UABS is set to 2 m, while its cruising speed to 1 m/s. The loitering time to collect user feedback is 2 s, allowing the UE to measure the metrics of interest and to attenuate non-deterministic aspects due to the drone hovering. The initial distance between UE and UABS is 9.5 m. In Fig. 4b, we see the path traveled by the drone, and registered by the ground station, while running the optimization algorithm. During each experiment, the UABS reached the global optimum in 11 minutes, on average, getting as close as 0.4 m from the UE. The average post-optimization measured throughput value is 9.97 Mbps (see Fig. 5), very close to the transmit throughput of 10 Mbps.

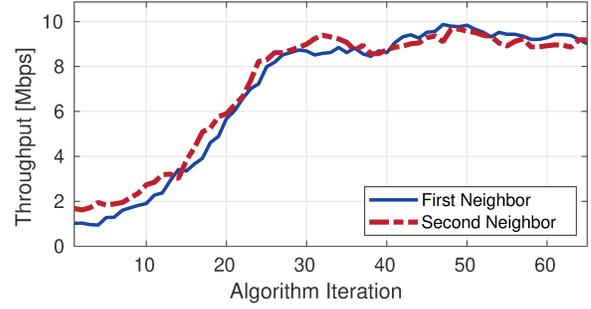


Fig. 5: Throughput convergence vs. optimization algorithm iterations.

Fig. 5 shows the throughput values received by the UABS as feedback from the UE during different iterations of the optimization algorithm. For the sake of clarity, only the curves of a single experiment are shown. During each iteration, the UABS moves toward two neighbor locations close to the position computed in the previous iteration: nominally *first* and *second neighbor*. We see that both neighbor curves start from a value between 1 and 2 Mbps and increase until they converge to approximately 9.5 Mbps. This shows that the optimization algorithm effectively manages to get very close to the global optimum of 10 Mbps.

In Fig. 6, we show the measured throughput by the UE relative to the downlink data sent by the UABS. Contrary to what done before, in this experiment we do not set any fixed transmit rate. We notice that the throughput starts with values as high as 15 Mbps and then decreases with the increase of the UABS-UE distance due to path loss. By leveraging this metric in the optimization algorithm, the UABS is able to effectively optimize its location and provide a better service to mobile subscribers (see Fig. 5).

VII. DISCUSSION

The provided results are limited to a single UABS and UE in a controlled environment. Further study is needed to fully understand additional factors, such as the height of the drone, the type of antennas involved, the angle between transmitter

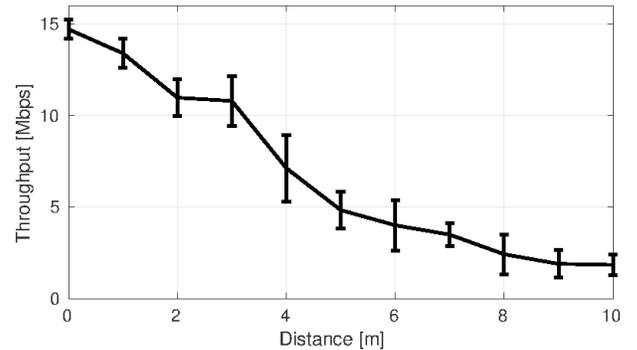


Fig. 6: Average throughput measured by the UE as a function of the UABS-UE distance, 95% confidence intervals are shown.

and receiver, and so on. Taking these parameters into account could improve the obtained optimization results at the expense of a higher computational complexity. This, though, would hardly be sustainable by compute boards with limited capabilities such as the ones not exceeding the payload of medium and small drones.

Thus, the current work represents a starting point in developing next generation mobile HetNets as a cost-effective solution to quickly recover from temporary failures of the terrestrial infrastructure. Moreover, the current work could be extended in a multi-cell scenario, in which several UABSs cooperate to reach a common goal.

The presented approach, while allowing UABSs to save computational resources, suffers from some issues related to use of greedy algorithms. These are, for instance, a tight dependence on the underlying objective function and the number of optimization steps required for convergence. The first issue is still an open research problem, since there is no one-fit-all solution, but solutions must be tailored for the each specific application, instead. On the contrary, the second issue could be mitigated by introducing memory capabilities in the gradient-based algorithm to prevent the UAV from visiting the same area multiple times in a short time-lapse. Another possible solution would be sharing the drones' knowledge to cooperatively map the area of interest.

VIII. CONCLUSIONS

We presented a quality-aware aerial-to-ground 5G cell framework with position optimization capabilities for next generation Unmanned Aerial Base Stations (UABSs). The proposed system aims at improving users' service in scenarios in which the underlying mobile infrastructure fails or temporarily overloads. The devised approach runs a gradient-based algorithm to compute the optimal UABS position relying on metrics of the UABS-UEs system only, and without using any additional source of information. In this way, our framework also manages to reduce the execution overhead and workload. We prototyped the devised UABS by leveraging standard-compliant 4G/5G software and off-the-shelf commercial hardware, and experimentally demonstrated its effectiveness. Results show that the UABS is able to find the optimal position to enhance service provided to mobile subscribers.

Future work will focus on further experiments involving a larger number of devices, on improving our algorithm robustness to wireless channel conditions (e.g., collisions and interference) in multi-UABS scenarios, and on developing multi-device interference-aware dynamic positioning algorithms.

REFERENCES

- [1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.
- [2] H. C. Nguyen, R. Amorim, J. Wigard, I. Z. Kovacs, and P. Mogensen, "Using LTE networks for UAV command and control link: A rural-area coverage analysis," in *Proc. of IEEE VTC-Fall*, Toronto, ON, Canada, Sept. 2017, pp. 1–6.
- [3] E. Greenberg and P. Levy, "Channel characteristics of UAV to ground links over multipath urban environments," in *Proc. of IEEE COMCAS*, Tel-Aviv, Israel, Jan. 2017, pp. 1–4.

- [4] R. de Silva and S. Rajasinghege, "Optimal desired trajectories of UAVs in private UAV networks," in *Proc. of IEEE ATC*, Ho Chi Minh City, Vietnam, Dec. 2018, pp. 310–314.
- [5] F. Cheng, S. Zhang, Z. Li, Y. Chen, N. Zhao, F. R. Yu, and V. C. M. Leung, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 7, pp. 6732–6736, July 2018.
- [6] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment algorithms of flying base stations: 5G and beyond with UAVs," *IEEE Internet of Things Journal*, 2019.
- [7] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3D drone placement in 5G+ wireless networks," in *Proc. of IEEE ICC Workshops*, Paris, France, July 2017, pp. 109–114.
- [8] C. Dixon and E. W. Frew, "Optimizing cascaded chains of unmanned aircraft acting as communication relays," *IEEE JSAC*, vol. 30, no. 5, pp. 883–898, June 2012.
- [9] J. Chakareski, S. Naqvi, N. Mastrorade, J. Xu, F. Afghah, and A. Razi, "An energy efficient framework for UAV-assisted millimeter wave 5G heterogeneous cellular networks," *IEEE Trans. on Green Communications and Networking*, vol. 3, no. 1, pp. 37–44, Mar. 2019.
- [10] R. Gangula, O. Esrafilian, D. Gesbert, C. Roux, F. Kaltenberger, and R. Knopp, "Flying rebots: First results on an autonomous UAV-based LTE relay using OpenAirinterface," in *Proc. of IEEE SPAWC*, Kalamata, Greece, June 2018.
- [11] P. Ladosz, H. Oh, and W.-H. Chen, "Optimal positioning of communication relay unmanned aerial vehicles in urban environments," in *Proc. of IEEE ICUS*, Arlington, VA, USA, July 2012, pp. 1140–1147.
- [12] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: Performance and optimization," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2068–2085, July 2011.
- [13] M. Gapeyenko, V. Petrov, D. Moltchanov, S. Andreev, N. Himayat, and Y. Koucheryav, "Flexible and reliable UAV-assisted backhaul operation in 5G mmWave cellular networks," *IEEE JSAC*, vol. 36, no. 11, pp. 2486–2496, Nov. 2018.
- [14] L. Bertizzolo, M. Polese, L. Bonati, A. Gosain, M. Zorzi, and T. Melodia, "mmBAC: Location-aided mmWave backhaul management for UAV-based aerial cells," in *Proc. of ACM mmNets*, Los Cabos, Mexico, Oct. 2019.
- [15] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, Nov. 2017.
- [16] S. Iellamo, J. J. Lehtomaki, and Z. Khan, "Placement of 5G drone base stations by data field clustering," in *Proc. of IEEE VTC Spring*, Sydney, NSW, Australia, Nov. 2017, pp. 1–5.
- [17] L. Ferranti, F. Cuomo, S. Colonnese, and T. Melodia, "Drone cellular networks: Enhancing the quality of experience of video streaming applications," *Elsevier Ad Hoc Networks Journal*, vol. 78, pp. 1–12, 2018.
- [18] M. Deruyck, J. Wyckmans, L. Martens, and W. Joseph, "Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario," in *Proc. of IEEE WiMob*, New York, NY, USA, Oct. 2016, pp. 1–7.
- [19] L. Ferranti, S. D'Oro, L. Bonati, E. Demirors, F. Cuomo, and T. Melodia, "HIRO-NET: Self-organized robotic mesh networking for Internet sharing in disaster scenarios," in *Proc. of IEEE WoWMoM*, Washington, D.C., USA, June 2019.
- [20] "OpenAirInterface," <https://www.openairinterface.org>, 2019.
- [21] "PX4 flightcontroller," <https://px4.io>, 2019.
- [22] "MAVLink protocol," <https://mavlink.io>, 2019.
- [23] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. on Automatic Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992.
- [24] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *arXiv preprint arXiv:1511.06807*, 2015.
- [25] D. C. Chin, "Comparative study of stochastic algorithms for system optimization based on gradient approximations," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 27, no. 2, pp. 244–249, Apr. 1997.
- [26] J. C. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 817–823, July 1998.
- [27] V. Fabian, "Stochastic approximation," in *Optimizing methods in statistics*. Elsevier, 1971, pp. 439–470.