UPFlight: An enabler for Avionic MEC in a drone-extended 5G mobile network

Umberto Fattore\textsuperscript{1,2}, Marco Liebsch\textsuperscript{1}, Carlos J. Bernardos\textsuperscript{2}
\textsuperscript{1}NEC Laboratories Europe GmbH - Heidelberg, Germany
\textsuperscript{2}Universidad Carlos III de Madrid - Madrid, Spain
Email: \{umberto.fattore — marco.liebsch\}@neclab.eu — cjbc@it.uc3m.es

Abstract—The interest in extending the range of mobile networks by means of drones, which carry a cellular radio base station or even a complete mobile core, is continuously increasing in research as well as in industry. Whereas the asset in such solutions is well motivated, various practical aspects are not fully considered, such as weight, power consumption and associated flight time of such drones, which usually carry limited and expensive mobile energy resources. In this paper we leverage various advantageous architectural and operational aspects of a 5G Mobile Core and propose the use of a low-cost system-on-chip solution on board of a drone to realize a flying 5G User Plane Function, which serves as data plane anchor to mobile devices attaching to the cellular base station co-located with the drone and enables offloading traffic to services, which can run on the system-on-chip board as well. We prototype a standard-conform User Plane Function by means of the open source fd.io project for Vector Packet Processing and enabled mobile access to flying lightweight mobile edge services. The paper describes how the drone system operates with the architecture backend in the fixed infrastructure and presents experimentally determined performance figures, comparing the developed low-costs flying edge services with a setup, which deploys the 5G User Plane Function as well as services in the fixed infrastructure.

I. INTRODUCTION

The ambitious and cloud-native design of the 5th Generation (5G) mobile communication system enables very customized and tailored deployment of the associated architecture's network functions. Contrary to the 4th Generation, named Evolved Packet Core (EPC), 5G clearly separates control- from data-plane functions and enables independent evolution and deployment, without even implying how a function is realized, i.e. in hardware, software or instantiated as virtualized network function (VNF). The 5G data plane function, which serves as a data plane anchor for a mobile device's traffic, is denoted as User Plane Function (UPF) and can be deployed in a decentralized way. To keep the topological and routing distance between a service and its user short, a UPFs can be deployed in edge network data centers in the proximity of the radio base stations being used by the mobile devices, as well as close to their correspondent service, which can be hosted in the same edge data center as the UPF. A platform for hosting services and applications in such network edge clouds is denoted as Multi-access Edge Computing (MEC) platform, which is specified in the ETSI MEC standards organization. MEC assumes traffic between its hosted service and a corresponding mobile device can be routed through a nearby UPF, as all traffic from and to a mobile device needs to traverse the device’s data plane anchor. Thanks to the availability of lightweight programmable hardware for software radio, the Universal Software Radio Platforms (USRP), cellular radio base stations can be realized with a small form factor, which makes them interesting for dedicated deployment, such as on drones. Recent research investigated the use of drones carrying cellular base stations to support a diversity of use cases, such as supporting mobile communication on-demand for white spots, which are not covered by the operator's infrastructure base stations, or hot spots, which require additional capacity or cellular coverage after failure of the infrastructure base stations. The research does not take specific assumptions on the location of the mobile core functions, such as the mobility data plane anchor or control plane functions for mobility management.

In this paper we investigate the use of a fast data plane Open Source Software for Vector Packet Processing (VPP) [1] on a very lightweight single-board computer to realize a low-power/low-cost/low-weight UPF, which can be deployed on a drone together with a lightweight radio base station. While the reduced weight and power consumption contributes to a prolonged flight time of the drone, such deployment allows moving the mobile edge and services even closer to users, namely on the drone, and enables most optimized data plane routes for the communication between a mobile device and a MEC service, or between two mobile devices being attached to the same or different drones without mandating the data plane traffic to traverse the network infrastructure. Exemplary applications can include voice- and video communication services or message brokers (e.g. AMQP). At the same time, keeping the remaining mobile core control plane and associated management and orchestration functions in the network infrastructure allows a centralized control and very flexible deployment of radio base stations along with co-located data plane nodes and (flying) edge services. In this paper we denote such proposed solution as Avionic MEC.

Whereas we see clear advantages of a mobile data plane realized in software and running in the user space of the flying lightweight board, such as ease in the configuration of an intrinsically available and rich mobile data plane protocol suite, the constrained compute board results in limitations of the effective KPIs associated with the mobile data plane. These constraints include the limited performance of the ARM architecture based processor, lack in hardware support for a fast data plane, such as Intel’s DPDK, or sharing a single USB controller to handle all network- and interface data\textsuperscript{1}.

\textsuperscript{1}Listed are characteristics per the specification of the single-board computer selected and used in our lab environment.
This paper first introduces background and related works in Sec.II and then motivates the use of a lightweight UPF on drones, presenting design objectives, functions’ deployment and operational aspects for three key communication scenarios in Sec.III. The implementation of a VPP-based mobile UPF on an ARM Cortex-A53 based single-board computer as well as its integration and deployment with our experimental 5G Core testbed using cellular LTE radio access is described in Sec.IV and some results are presented in Sec.V.

II. BACKGROUND AND RELATED WORKS

Latest specified 3rd Generation Partnership Project’s (3GPP) 5G Core (5GC) architecture [2] and procedures [3] enable a highly flexible deployment of functions both in the control plane (leveraging the principles of the Service-Based Architecture - SBA) and in the data plane. Here, one or more UPF(s) are meant to compose the data path, with the last one acting as PDU Session Anchor (PSA) for all user traffic, and others as Uplink Classifiers (UP CL) to be used as branching points, e.g. to access a local data network. A simplified architecture is depicted in Fig. 1. Main control plane functions are the Access and Mobility Management Function (AMF) and the Session Management Function (SMF), taking care of user authentication and session management, respectively. The interface between the Radio Access Network (RAN) and the first UPF is named by 3GPP as N3, and here packets are encapsulated using the GRE Tunnelling Protocol for User Traffic (GTP-U) [4]. Following, N9 interface lays between UPF(s) and N6 between the last UPF and the Data Network (DN) where services are deployed. N9 is still meant to have encapsulated traffic, while N6 is actually considered out of scope from 3GPP and so no specific protocol is specified to be used. N2 and N4 interfaces are used between the user plane and the AMF and the SMF, respectively.

In the current state of the art, several works highlight the advantages (and the challenges) of base stations deployed on flying devices (i.e., drones) in order to achieve flexible deployments for next generation mobile network. The feasibility of such scenario, based on current standards, is discussed in [5] while [6] provides drone-to-ground radio connectivity performance evaluation based on some drone flight parameters, such as height and speed. Different trials were conducted deploying LTE base stations on Unmanned Aerial Vehicles (UAVs), for instance the ones by AT&T [7] and Verizon [8].

All the cited works mainly focus on the radio components, assuming the mobile core deployed on the ground (e.g., in a Data Center or even in a van or vehicle deployed in the drones flying area proximity). This introduces on one side the challenge of the radio-to-mobile-connectivity (i.e., forcing for the use of a less reliable wireless channel or for a wired cabling, but then strongly reducing the mobility of the drone); and on the other side, this discourages any data plane optimization, forcing user traffic to always traverse the on ground infrastructure. A first analytic work considering edge deployment on UAVs in order to minimize the total mobile energy consumption and offload mobile application, while satisfying quality of service requirements, is presented in [9]. More, [10] presents an architecture in which the entire mobile core is deployed on the flying device: this enables edge computing (with the possibility to even have the entire data path not going beyond the drone), but anyway is not devoid from drawbacks, among all the exacerbation of the inter-core communication complexity and the higher power consumption because of the high number of functions deployed on board. [10] also focuses on the additional procedures which need to be introduced because of inter-core communication, allowing for an optimized mobility scheme but loosing compatibility with current 3GPP’s mobile core standards. [11] addresses the optimization of a UAV’s trajectory under consideration of limited mobile energy resources, taking the UAV’s energy consumption for propulsion as well as for wireless communication into account. Other works show the advantages of edge computing deployments in order to enhance mobile user performances (i.e., in particular latency), while maintaining strong compliance to 3GPP’s architecture [12], thus not focusing specifically on drone deployments.

In our solution, we equally convey on the necessity for having some mobile core functions deployed on board the flying device, in order to enable edge computing capabilities and enhance user plane performances; but at the same time we want to be selective in which functions deploying on board and deploy the remaining (i.e., less fundamental for the user plane optimization) on the ground infrastructure. This is summarized in our concept of a UPFlight, offloading the drone both in terms of weight and power consumption.

III. THE LIGHTWEIGHT UPF IN THE AIR

A. Motivation and Design Objectives

Motivations beyond the introduction of drones in mobile core architectures mainly involve support in case of natural disasters [13] or service provisioning in highly mutating scenarios (e.g., in massive events) [14]. Deploying user plane functions at the edge (i.e., the avionic edge, on the drone) introduces some additional advantages, such as: (1) enabling optimized user plane paths (i.e., avoiding passing through the on-ground-infrastructure when the communication occurs, for instance, between two devices attached to the same or different drones); (2) achieving lower delay for users accessing services on the drone, and (3) reducing load on the wireless link which connects the drone with the infrastructure.

A drone deployment brings some new challenges, both concerning limited sustainable weight and power supply on

![Diagram of 5G Core user plane functions and interfaces in the user plane and towards control plane functions.][1]

[1]: Fig. 1. 3GPP’s 5G Core (5GC) user plane functions and interfaces in the user plane and towards control plane functions. [2]
board of the drone and overload of the channels connecting to the ground. This mainly marks the point of deciding where to deploy each mobile core function (and on which kind of device) to ensure an acceptable balance in addressing various, sometimes contradicting objectives. In our solution we deploy 3GPP’s 5G mobile core functions (and introduce additional orchestration level and drone management functions), focusing on 3 fundamental design objectives:

1) to minimize the traffic load on the wireless links between the drone and the ground infrastructure;
2) to minimize power consumption and weight load on board of the drone;
3) to enable MEC on board of the drone, bringing (when possible) the service near to the mobile user, achieving enhanced user performance and core offloading.

A trade-off is necessary in considering the above objectives, since they often conflict with each other: e.g., minimizing on board power consumption means offloading from the drone as many functions as possible, thus exacerbating the load of the channel connecting flying deployed functions to those on the ground. Keeping offloaded the communication between on board and on ground infrastructure is important since the wireless link in between may be unreliable and therefore implies performance degradation of user services, as also shown in the experimental results discussed later in this paper.

**B. UPFlight Deployment**

Figure 2 depicts the proposed solution, highlighting the functions which are deployed on board of the flying device and those in the Data Center. Some additional elements are introduced in order to synchronize the UAV* management (automated positioning, monitoring of system metrics - e.g., remaining battery power) and mobile network control components.

On board of the UAV, the base station and 3GPP’s User Plane Function (UPF) are deployed. Positioning the UPF on board of the device is the fundamental enabler towards avionic MEC, since user traffic needs to traverse at least one UPF node deployed along the data path. With this setup, we minimize the distance covered by 3GPP’s N9 interface, mitigating the drawbacks given by the use of the GTP-U encapsulation protocol [15]. This brings the data plane anchor associated with the user’s data session towards the edge, in order to then have maximum flexibility in the protocols and routing for the second part of the user traffic path between the UPF and the service’s data network (N6 interface), leveraging the advantages already described and proved in [12]. Since 3GPP does not specify details on how to treat data plane traffic on the N6 interface, we co-locate a programmable Data Plane Node (DPN) function with the UPF to enforce uplink traffic steering rules for flexible routing of data traffic either to the on-board Data Network (Flying DN in Fig. 2) or to the Data Center Data Network, or even to a different DPN deployed on another drone (e.g. for optimized communications between mobile users attached to different UAVs).

We deploy the control plane components on the ground Data Center, in order to avoid overloading the flying device with many functions (exacerbating its power consumption) and to centralize the control of the drone-extended 5G mobile network, i.e. flying UPF placement and selection. Doing this, we accept loading the wireless link connecting the UAV to the ground with control plane signalling, a price we decide to pay in order to reach a trade-off among our before listed design objectives. What is offloaded from the wireless link is all user plane traffic directed to the Flying DN, which is potentially higher than signalling control used only for initially creating the data path and manage it when modifications are required (e.g. in case of handover). Finally, having a centralized control-plane among many UAVs simplifies the management of handover procedures and user mobility between drones, whereas specific alternative solutions should be designed in case a control plane is deployed on board of each drone. Going more in depth into the control plane components, these may be distinguished between those composing the Mobile Network Control Plane (C-PLANE), responsible for the management of mobile user sessions (i.e. forwarding, traffic steering, authentication, etc.) and the UAV Controller, which is responsible for the drone behavior (e.g., speed, positioning). Each drone has a UAV Agent which monitors parameters from the drone and processes them locally, raising alerts towards the UAV Controller when some threshold value is reached. The UAV Controller may request for specific parameters and enforce rules to the UAV Agents (e.g., enforce rule for changing the drone positioning). For instance, the UAV Agent collects data concerning the drone remaining battery and, when this is under a certain percentage, raises an alert towards the UAV Controller.

The above Orchestrator entity is then needed to synchronize the orchestration of resources and functions for connectivity with the results of drone-related data monitoring. This Orchestrator collects data from the UAV Controller and the Mobile
Network C-PLANE and takes decisions to move UAVs (e.g., to rearrange a better coverage of the area or to send a drone for battery recharging) or to relocate mobile network functions.

Concerning the Mobile Network C-PLANE, next to the 3GPP’s Service Based Architecture (SBA) defined in [2], we introduce the N6 Controller, which takes care of enforcing traffic treatment policies on the DPNs, e.g., for traffic steering on the N6 interface. The integration of this additional control element to the 3GPP’s domain leverages the use of 3GPP’s Application Function (AF), allowing for inter-working with additional external functions. This allows (with request-response and subscription-notification alike mechanism) to receive updates from the Session Management Function (SMF) about changes in the configuration of a user’s UPF in the 3GPP’s domain and also to initiate such updates. However, details about this inter-working between 3GPP’s and non control elements is out of the scope of this paper.

Our solution assumes the utilization of a wireless channel from the UAV to the ground infrastructure for multiple signalling interfaces (i.e., 3GPP’s N4 and N2 and signalling interface for the on-board DPN) and for user plane traffic in all those scenarios in which the service is not on-board deployed (i.e., N6 interface, according to 3GPP terminology). The wireless technology used for this channel should strongly consider reliability metrics in order to avoid frequent interruptions during the signalling procedures or user sessions. Indeed also the UAV management signalling uses a wireless channel, which could both be shared with all the mobile core functions or independent.

Finally, an important aspect we pointed out at the beginning of this section is related to power constraints. In order to achieve this, both the radio components and the physical device on which core functions are deployed should be able to ensure low power consumption and be limited in weight as well. Further in this paper, this aspects are strongly considered by evaluating our system using low-cost system-on-chip solutions at this scope.

C. Operational Details

Figure 3 depicts some scenarios on which we focus in a drone-extended mobile network deployment, describing high level behaviors from the main architectural components and the optimized user paths, which our solution enables.

1) Leveraging the Service Request Procedure (a): Basic scenarios include the mobile user requesting access to a specific service, which can be deployed both on the Flying Data Network or on the Data Center infrastructure. The Orchestrator keeps track of the deployment of each service and may also decide for its relocation based on the amount of requests and on the network monitoring. The designed procedure assumes to happen beyond the regular 3GPP Registration Procedure for mobile user registration [3] (which involves only the 3GPP control plane functions, i.e., AMF, SMF, UDM, etc.) and leverages the 3GPP Service Request Procedure per [3] in which the mobile user requests for one (or more) service(s). The main innovation is the introduction of the Orchestrator in the procedure. The compliance to standards is ensured through the use of 3GPP’s AF, which allows for 3GPP’s external entities to request updates and suggest changes in the user plane data path (using standard defined Traffic Influence messages [3]). Therefore, when the mobile user requests a service (1a), this request reaches the 3GPP’s C-PLANE through the radio access (2a); the Orchestrator is updated (3a) on the service requested and checks whatever the service is available on the Flying DN and influences the 3GPP’s C-PLANE decisions accordingly. The 3GPP’s C-PLANE instructs the user plane functions on the path (4a) to reach the service locally or in the data center.

2) Mobile-to-Mobile Communication: A User Plane Function deployed at the edge of our infrastructure (i.e., on-board of the UAV) leverages the communication between different mobile users which are attached to different antennas on different UAVs, because it allows a direct communication without the necessity to pass through the on-ground infrastructure. This is a common situation in both big events (e.g., two participants want to exchange data the one with the other) or in emergency situations (e.g., allowing smoother communications...
among rescuers). The user requests for the connection through radio antenna and the request is forwarded to the core control functions (1b). Again, the Orchestrator receives updates of mobile users requests from the 3GPP’s C-PLANE (2b) and based on its knowledge on the network, it gives indications for a specific optimized path, which is then again enforced by the SMF to the involved UPF(s) on both drones (3b).

3) Drone Relocation: Drones have limited battery capacity and therefore limited flight autonomy. The average Flight Time (i.e., the maximum amount of time the drone is able to fly before needing battery recharge) of a regular drone is around 30 minutes [16], but it strongly depends on different factors (discussed in further sections). Therefore, the drone must return on the ground for recharging every while, which requires mechanisms for drone relocation, meaning the relocation of the functions and session states on board of the drone to a substitute drone. In doing this, session continuity is a fundamental achievement to avoid the user session to be interrupted.

From the mobile network point of view, this means (1) forcing a handover for all the users connected to the drone, (2) changing the user plane path in the mobile core and (3) migrating the service and the service state if this was deployed on the Flying DN. Once again, this happens through the Orchestrator, which has both the control of drone parameters (i.e., including battery power) and of mobile core functions and configurations. The on-board UAV Agent function monitors parameters of the drone and raises an alert towards the UAV Controller (1c) when the battery power is low. This is forwarded to the Orchestrator (2c), which finds a suitable substitution for the drone, choosing one or more drones (eventually enforcing changes in their position through the UAV Controller - 3c). The Orchestrator leverages the already described Traffic Influence messages [3] to force a mobile core triggered handover (4c), in order to have all users, which are attached to the exhausted drone, to change to the new one. The forwarding rules on the user plane elements on board of the drone are changed or deleted respectively (6c).

To enable user session continuity, the Orchestrator may also enforce the service state to be migrated from the previous drone to the new one (5c). In deciding this, the Orchestrator may take into account different factors to determine if the migration of the service state is affordable in the current situation of the network. In any case, the migration needs some time to be completed: if the handover was already accomplished from the 3GPP’s mobile core to the functions on the new drone, DPNs and N6 traffic steering rules (7c) can be used to initially define a sub-optimal route to the previous drone (2*) until the service’s state on the new drone is completed and usable (3*).

IV. EXPERIMENTAL SETUP

A. Methodology

In following sections we present the developed testbed, as a proof of feasibility and to highlight advantages of a light UPF co-deployed on the drone with the base station and a wireless link to connect to an on-ground infrastructure carrying other core components. We use a Raspberry Pi 3 B+ to host our edge User Plane Function (UPF). We make evaluation first comparing the Raspberry deployment with a classical Server deployment and then discuss some results when testing mobile user delay and throughput in an edge and in a Data Center (DC) deployment, considering both Mobile to Service (M2S) and Mobile to Mobile (M2M) communication.

B. SoC-based UPFlight - Integrated View

In Figure 4 the experimental setup is depicted, with a Raspberry Pi and a USRP base station deployed on board of the drone (DJI Matrice 600 PRO model), connected through a wireless connection with a server machine. On the server machine, the control plane functions are deployed as two virtual machines on a VMWare Workstation platform4, respectively consisting in: (1) an MME prime (MME’) taking care of Non-Access Stratum (NAS) protocol towards the LTE antenna and managing mobile user authentication (similarly to a 3GPP’s AMF); and (2) a Session Manager (SM), managing the user sessions and instructing the user plane nodes (similarly to a 3GPP’s SMF). Both AMF and SM were derived from OpenEPC rel. 75 software components as described in [12]. An additional virtual machine is used for deploying the service (SRV) allocated on the server machine; another service application runs on the Raspberry Pi as an edge service.

The User Plane Function (UPF) is deployed on a Raspberry Pi 3 B+ and is implemented using Fd.io Vector Packet Processing (VPP) software6, to enable user plane traffic fast

---

4VMWare Workstation - https://www.vmware.com/products/workstation-pro.html
5Core Network Dynamics (CDN) - https://www.corenetdynamics.com/products
6Fd.io Vector Packet Processing (VPP) - https://fd.io/technology/
forwarding and GTP-U support. An additional UPF is deployed on the Infrastructure Server for comparison, as described further in the experimental results section, implemented using software components from srsLTE software.\(^7\) Packets arriving on the Raspberry are forwarded through ovs-switches\(^8\) either to the VPP user space to act as an UPF (here they are encapsulated in downlink or decapsulated in uplink) or to the the Infrastructure Server (i.e., GTP-C for the control plane, GTP-U for the non-edge deployed UPF).

The radio base station and the Raspberry are wired connected with 10G Ethernet, while the connection towards the Infrastructure Server uses WiFi 802.11g in Infrastructure mode, using the Raspberry’s on board WiFi and a Netgear WG602v3 access point for the server.

V. EXPERIMENTAL RESULTS

The use of a constrained and low power device such as a Raspberry Pi implies limitations in the performance (i.e., computational power, bandwidth, etc.) which can be directly observed in the measured data rate values. Referring to this metric, Fig. 5(a) depicts a comparison between the Raspberry Pi used in our testbed and a server machine equipped with 32 cores and 64GB of RAM equipped with the same UPF VPP-based implementation. Plain traffic is generated and sent to the Raspberry/Server via the N3 up-link GTP-U tunnel and resulting encapsulated traffic evaluated to obtain data rates. Higher server performances are evident from 5(a), with the Raspberry setup achieving limited data rates before significant packet loss arises. Nevertheless, the Raspberry performance is found to be good enough considering its expected edge positioning and therefore the lower amount of users expected to be served, if compared to a server deployed centrally in the network.

Moreover, a server machine on board of the drone has a strong impact on the power consumption of the drone itself, therefore reducing the flight autonomy. A low Flight Time ends up in the need for frequent recharging and therefore frequent function relocation, exacerbating the mechanism described in the operational section (Fig. 3(c)). An estimation of Flight Time based on the weight on board and on the device consumption was made applying the following formula:\(^9\)

\[
FT = 60 \frac{(\text{BatteryCapacity} \times 0.8) - \text{DeviceConsumption}}{\text{AUW}/\text{TrustedWeightForAmpere}}
\]

where the All Up Weight (AUW) [Kg] is the total weight on board (including the drone itself), and TrustedWeightForAmpere [Kg/A] represents the trusted weight the drone is able to raise for each ampere (this parameter depends on the drone’s motors), BatteryCapacity [Ah] represents the overall capacity of the battery on board of the drone (a safe 0.8 multiplier is considered), while DeviceConsumption the consumption of all the devices deployed on the drone [Ah]. Fig. 5(b) depicts a simulation of Flight Time varying the AUW parameter, considering consumption for a Raspberry or for a server machine both while under high traffic load and not. Power Consumption for the Raspberry was calculated according to [17], considering a CPU load of 0.2 and 0.8 and a rate on both WiFi and Ethernet interfaces of 20 Mbit/s and 80 Mbit/s for the the low-performance and high-performance pattern respectively. For the server, average value from [18] were used, from idle and hdparsbench.\(^10\) A BatteryCapacity of 10 [Ah] and a TrustWeightForAmpere of 0.2 [Kg/A] was considered. The BatteryCapacity chosen was considered reasonably realistic based on the fact that the drone model actually used (and depicted in Fig. 4), is equipped of six batteries of 4.5 [Ah] each. Assuming an equal weight (which, of course, is not true), Flying Time ensured using a Raspberry can also be 100% more than the desktop setup (200% considering the high-performance case). Taking in account also the weight difference this trend is even more evident.

Moving to user performances, Fig. 6 depicts user plane throughput and delay measurements obtained in a laboratory setup using smartphone devices and a commercial NEC Orion base station for LTE radio access. Fig. 6(a) compares downlink throughput results measured using iperf tool\(^11\) from a mobile user (acting as iperf client) connecting to a service (SRV in Fig. 4, where iperf was running as a server) deployed respectively on the Raspberry Pi (i.e., the Edge) and on the server machine beyond WiFi (i.e. the Data Center - DC). Throughput rates are in average 25% better in the Edge deployment (in average, 10.7 against 7.9 Mbit/s) and much less variable (standard deviation of 0.36 against 3.5 - i.e., ten times less). This instability (and low performance) is introduced by the WiFi channel which in the DC service deployment is traversed by data traffic.

Outperforming of the Edge deployment is even more evident in the delay measurements depicted in Fig. 6(b). These are Round Trip Time (RTT) delays measured using the ping tool, considering a Mobile to Service (M2S) and a Mobile to Mobile (M2M) communication. In the first, the target address for the ping tool is the service (SRV) either located on the Raspberry (Edge) or on the server (DC); in the latter case, the target is the address of another mobile phone attached

---

\(^1\) srsLTE - https://www.srsoftwareradiosystems.com/tag/srslte/
\(^2\) Open vSwitch (OvS) - https://www.openvswitch.org/
\(^3\) This equation was derived from the one used for sizing weight on a drone - http://www.rcdrongood.com/quadcopter-battery-calculator-flight-times.

\(^4\) http://www.rcdronegood.com/quadcopter-battery-calculator-flight-times.

\(^5\) The hdpar benchmark from [18] was considered as highest consumption case since in the cpburn case - even more consuming - the estimated flight time would be less than zero.

\(^6\) Iperf - https://iperf.fr/
to the mobile network. When compared with Data Center (DC) deployment (i.e., UPF deployed beyond the WiFi on the Infrastructure Server) delay is significantly higher than in the correspondent Edge deployment (i.e., UPF and SRV both on the Raspberry). In the DC-M2S case, delays are about three times higher than in the Edge-M2S case, while in the DC-M2M case they are about double of the Edge-M2M scenario. These results are once more caused by the WiFi channel used in the DC deployments: in the Edge-M2S scenario, in fact, the radio channel is traversed twice (round trip) by user traffic and the WiFi channel none; in the DC-M2S case, both radio channel and WiFi channel are traversed twice. Very similarly, in the Edge-M2M only the radio channel is traversed (four times), while in the DC-M2M case the WiFi channel is traversed four times as well! Delay variability is higher every time an higher number of times the WiFi channel is traversed. As a note on absolute RTT delay values, the experiments were done using an LTE base station, which still brings in account about 20/25 ms delay for traversing the radio channel (40/50 ms for RTT). Therefore, the advantages of an Edge deployment would be even more significant with a 5G New Radio base station.

VI. CONCLUSION AND FUTURE WORKS

In this work, we present a drone-extended mobile core architecture enabling local breakout and edge computing on board of drones (i.e., avionic MEC) in order to improve performance for users, optimize routes in the mobile core and offload traffic from drones-to-ground-infrastructure wireless links. We show the importance of a light UPF in order to allow the deployment on board of drones without affecting power consumption and flying autonomy. The testbed we implemented gives a proof of some of the architectural choices and of the advantages of edge computing in such scenario.

Future work includes the definition of decisions algorithms for the Orchestration entity, specifically to determine the best deployment for a service (i.e., at (one of) the flying edges or on the ground infrastructure) and leveraging the drone relocation mechanisms to cope with the limited flight autonomy.

ACKNOWLEDGMENT

The research leading to these results has been partially supported by H2020-MSCA-ITN-2016 framework under grant agreement No. 722788 (SPOTLIGHT) and H2020-ICT-2018 under grant agreement No. 825012 (5G CARMEN).

REFERENCES

[1] L. Linguaglossa, D. Rossi, D. Barach, D. Marjon, and P. Piester, “High-speed software data plane via vectorized packet processing,” Tech report, 2017.
[2] 3GPP, “System Architecture for the 5G System (Release 16),” 3rd Generation Partnership Project (3GPP), TS 23.501, March 2019.
[3] ——, “Procedures for the 5G System (Release 16),” 3rd Generation Partnership Project (3GPP), TS 23.502, March 2019.
[4] ——, “General Packet Radio System (GPRS) Tunneling Protocol User Plane (GTPv1-U),” 3rd Generation Partnership Project (3GPP), TS 29.281, Mar 2017.
[5] I. Bor-Yaliniz, M. Salem, G. Senerath, and H. Yankomeroglu, “Is 5g ready for drones: A look into contemporary and prospective wireless networks from a standardization perspective,” IEEE Wireless Communications, vol. 26, no. 1, pp. 18–27, 2019.
[6] X. Lin, V. Vajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H.-L. Maaitanen, M. Bergstrom, S. Euler, and Y.-P. E. Wang, “The sky is not the limit: LTE for unmanned aerial vehicles,” IEEE Communications Magazine, vol. 56, no. 4, pp. 204–210, 2018.
[7] Att, “When COWs Fly: ATT Sending LTE Signals from Drones,” bit.ly/ATTCOWs, 2017, [Online; accessed on April 2019].
[8] Verizon, “Verizon trials drones as flying cell towers to plug holes in internet coverage,” bit.ly/VruZoNvtC, 2016, [Online; accessed on April 2019].
[9] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a unmanned cloudlet: Optimization of bit allocation and path planning,” IEEE Transactions on Vehicular Technology, vol. 67, no. 3, pp. 2049–2063, 2018.
[10] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, “Skycore: Moving core to the edge for unthethered and reliable uav-based lte networks,” in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. ACM, 2018, pp. 35–49.
[11] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, “Energy trade-off in ground-to-uav communication via trajectory design,” IEEE Transactions on Vehicular Technology, vol. PP, 09 2017.
[12] U. Fattore, F. Giust, and M. Liebsch, “5gc+: An experimental proof of a programmable mobile core for 5g,” IEEE 23rd Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2018.
[13] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, “Drone-aided communication as a key enabler for 5g and resilient public safety networks,” IEEE Communications Magazine, vol. 56, no. 1, pp. 36–42, 2018.
[14] I. Bor-Yaliniz and H. Yankomeroglu, “The new frontier in ran heterogeneity: Multi-tier drone-cells,” IEEE Communications Magazine, vol. 54, no. 11, pp. 48–55, 2016.
[15] C. Saksik and S. K. M., “Sdn based evolved packet core architecture for efficient user mobility support,” in Network Softwarization (NetSoft), 2015 1st IEEE Conference on. IEEE, 2015, pp. 1–5.
[16] M. Hassanalai and A. Abdelkefi, “Classifications, applications, and design challenges of drones: A review,” Progress in Aerospace Sciences, vol. 91, pp. 99–131, 2017.
[17] F. Kaup, P. Gottschling, and D. Hausheer, “Powerpi: Measuring and modeling the power consumption of the raspberry pi,” in 39th Annual IEEE Conference on Local Computer Networks. IEEE, 2014, pp. 236–243.
[18] M. el Mehdi Diouri, M. F. Dolz, O. Glück, L. Lefèvre, P. Alonso, S. Catalán, R. Mayo, and E. S. Quintana-Oritz, “Solving some mysteries in power monitoring of servers: Take care of your wattmeters?” in EE-LIDS, 2013.