Design of an On-demand Agile 5G Multi-Access Edge Computing platform using Aerial Vehicles

Fabrizio Granelli , Università degli Studi di Trento

Cristina Costa, Fondazione Bruno Kessler

Jiajing Zhang, Technische Universität Dresden and Centre for Tactile Internet with Human-in-the-Loop (CeTI)

Riccardo Bassoli, Technische Universität Dresden

Frank H.P. Fitzek, Technische Universität Dresden and Centre for Tactile Internet with Human-in-the-Loop (CeTI)

Fast on-demand 5G connectivity can be deployed through the usage of Aerial Platforms. Indeed, the usage of moving nodes represents at the moment the most interesting and cost-affordable way to bring connectivity and network services in emergency scenarios or in case of absence of the network infrastructure. This paper presents an architecture for using drones as movable base stations, interconnected with a high altitude platform, capable of deploying a Multi-Access Edge Computing following current ETSI standards. Moreover, a reinforcement learning algorithm is proposed to enable proper resource allocation in order to guarantee QoS requirements.

Introduction

Unmanned Aerial Vehicles (UAVs), especially drones, are increasingly being used by civil and military applications due to their low cost, agility and freedom of movement. However, while commonly vehicles, e.g. cars, are typically considered end-hosts or relay nodes of networks, UAVs can provide novel applications, e.g. in terms of swarms of mobile base stations as well as nodes in a mobile cloud, thus allowing on-demand delivery of connectivity and processing power. This is made possible by the fact that a UAV is typically capable of bringing a payload whose weight is approximately that of the drone itself - opening the possibility to equip drones with cameras, access points and small computing devices, such as the Raspberry PI or similar.

Modern mobile networks provide an unprecedented opportunity due to their flexibility and UAVs could represent a key deployment technology. Indeed, the 3rd Generation Partnership Project (3GPP) 5G networks are expected to guarantee short deployment time, in the order of 90 minutes or less, thus opening the way for agile deployments in areas with limited coverage or network disruption. Such requirement, jointly with the expected flexibility deriving from the implementation of the 5G Service Based Architecture in the core and functional splits in the architecture of the base station, makes it possible to integrate UAV platforms.

To address the above scenario, and for example in emergency situations or rural areas, drones might be used as mobile base stations (BSs), *bringing the network to the user*. However, it is possible to even go beyond such concept and include processing capabilities in the picture, leading to the deployment of agile fog/edge computing solutions by *bringing the cloud to the user* through high-altitude platform (HAP) drones, delivering backhaul/fronthaul connectivity and services.

This concept is also defined as the Edge Computing (EC) paradigm, which focuses on deploying dynamic services capable of *following* their users. EC guarantees lower probability of congestion, higher performance and lower latency. However, it requires the definition of proper standards and operating environments in order to achieve wide deployment and integration within mobile networks.

ETSI standardization represents a relevant action to bring EC technology to 3GPP mobile networks in the form of the so-called Multi-Access Edge Computing (MEC) framework. MEC standards by the European Telecommunications Standards Institute (ETSI) provide the framework to deploy MEC solutions at the edge of the network, but they leave freedom on how to specifically implement those functionalities and to satisfy quality-of-service (QoS) requirements. Indeed, the operation time scale of QoS-aware resource allocation falls out of the possibility of human intervention, and thus it requires automation, through the introduction of proper Artificial Intelligence or Machine Learning solutions to identify the most appropriate allocation capable of satisfying users' QoS while maintaining high utilization of the available resources.

This paper proposes an architecture compatible with ETSI/3GPP standardization, aimed at the actual deployment of a flexible and automated MEC solution for deploying connectivity and services using a swarm of drones. Particularly, mobile BSs work in cooperation with a high-altitude platform (HAP) (e.g. a balloon or ultra-light vehicles). Basically, the drones build the radio access network (RAN) and the edge, providing robust and flexible communications, thus representing the programmable network infrastructure used to connect users and deploy services in an on-demand framework. Additionally, the article analyzes the deployment of network slicing to enhance assignment of resources. In particular, automation through different algorithms is implemented within the Slice Broker, showing the potential of reinforcement learning towards flexibility and satisfaction of QoS (e.g. transmission rate and latency).

Overview of Mobile Edge Cloud

Support for EC is considered by 3GPP a critical enabler of efficient service delivery in mobile networks, being able to provide reduced end-to-end latency and decreased load on the transport network [1]. 5G and beyond networks can be considered the natural future environment for MEC deployments, since many innovation aspects brought by 5G are indeed centered on applications, which are expected to be highly heterogeneous with often contradictory requirements [2].

With the purpose of supporting the EC adoption, ETSI has started around 2013 the MEC industry specification group (ISG). Initially known as Mobile Edge Computing, the definition of MEC has been changed to Multi-Access Edge Computing in order to accommodate a broader scope. The ISG core objective focuses on creating standards that support the cooperation between network operators, applications, and content providers. The final goal is to boost the overall QoS experienced by the user equipment (UE), while optimizing the benefits for all players. Currently, both ETSI ISG MEC and 3GPP are working on bringing Edge Computing into 5G and beyond architectures. Recently, the 3GPP SA6 group started to work on Edge Computing as an enabler for new vertical applications, and 3GPP plans to integrate MEC into 5G in the future release 17 [3]. Nevertheless, progress is still needed to fully integrate MEC into 5G and beyond architectures.

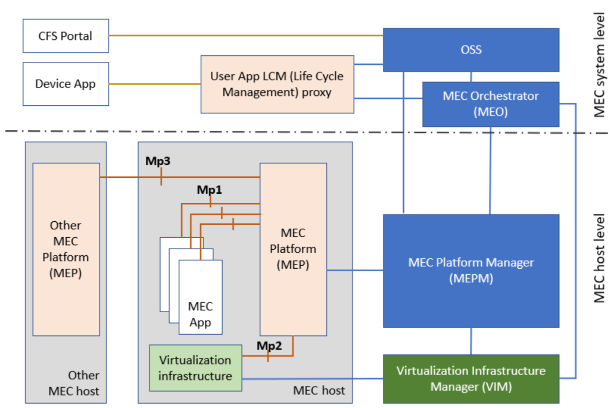

Figure 1 *Simplified ETSI MEC reference architecture (based on [1]).*

The ETSI MEC reference architecture [4] (see Figure 1), now in its second version, includes all the functional entities that characterize the MEC and the interfaces between them. Functional entities are divided into two groups:

- *MEC system-level entities*, e.g. the user app LCM proxy and the Mobile Edge Orchestrator (MEO), typically deployed in the core network;
- *MEC host level entities*, e.g. the mobile edge applications and the Mobile edge platform, typically deployed close to the UEs.

The device application residing on the UE leverages the ETSI MEC platform to interact with ME applications and services residing on edge nodes. These can also be co-located with base stations of the cellular network.

The Edge architecture consists of a ME Platform Manager (MEPM) and single to multiple MEC Host Sites. The MECM is a server centrally deployed, e.g. in a centralized cloud, which controls the deployment of applications on MEC Hosts. The MECM node can be e.g. a Kubernetes Cluster.

The Mobile Edge Orchestrator (MEO) is responsible for on-boarding and enabling the Edge Application required by the UE. MEO also chooses the optimal MEC host on which to deploy the application. This choice may be based on various parameters, such as latency, available resources, number of users and available services.

MEC Service APIs	Refenrece	Description
Application enablement API	ETSI GS MEC0011	Service-related functionalities.
Radio Network Information API	ETSI GS MEC012	Wireless network status information exposure
Location API	ETSI GS MEC013	Location information of the terminal exposure.
UE Identity API	ETSI GS MEC014	Allows registering a tag (ID) for the user's equipment (UE) to enforce traffic rules for that specific UE.
Bandwidth Management API	ETSI GS MEC015	Edge applications running at the same time on the same edge host can send to the MEP their bandwidth requirements.
Device App API	ETSI GS MEC016	Lifecycle management of the UE client application
WLAN Info API	ETSI GS MEC028	WLAN Access Information exposure
Fixed API	ETSI GS MEC029	Fixed Access Information exposure

Table 1 *A set of APIs for MEC apps.*

The traffic from the UE is steered so that it can reach the appropriate Edge Application when needed and processed locally. The User Plane Function (UPF) is a fundamental component of a 3GPP 5G core infrastructure system architecture [1] and is the function in charge of the routing of the user plane traffic to the appropriate Data Network

(DN), thus providing the encapsulation and de-capsulation of GPRS Tunneling Protocol for the user plane (GTP-U). In MEC, the UPF requires to be configured with appropriate traffic steering policies able to redirect traffic to the appropriate MEC applications [5].

The UPF configuration is triggered by the MEP, which, supported by the information received by the MEO during the application onboarding, interacts with the Mobile Core. In the 5G architecture, the MEP is expected to be integrated as a 5G AF [5]. The MEP may request the redirection of the UE traffic to a MEC application as per the request of the MEO via the MEPM.

In order to fully take advantage of the locality of the edge, ETSI has defined a set of vendor-independent REST APIs, as standardized interfaces, that allow the vendors of user devices/applications and edge node applications to interoperate. A "MEC application enablement framework" has been defined by ETSI to supporting the network exposing information towards authorized third-party applications. ETSI has delivered seven Group Specifications (Table 1) that define different REST multi-access edge service APIs addressing various aspects that range from mobile edge service APIs to application lifecycle management. The information are made accessible to Edge Applications developers through the Mp1 interface, that allows accessing to data either provided by the MEC platform itself, such as Radio Network Information (RNI API [6]) or location information (Location API [7]), and those provided as a services by other MEC applications. Those data services can be consumed not just by user-level applications, but also by services enabling network performance and QoS improvements.

In particular, the Radio Network Information (RNI) provides up-to-date radio network information to mobile edge applications and to mobile edge platforms. Typical information that may be provided is, e.g., up-to-date radio network information regarding radio network conditions; measurement information related to the user plane based on 3GPP specifications; information about UEs connected to the radio node(s) associated with the mobile edge host, their UE context and the related radio access bearers; changes on information related to UEs connected to the radio node(s) associated with the mobile edge host, their UE context and the related radio access bearers [6]. The information can be accessed with different granularities, depending if the data is requested e.g per cell, per User Equipment, per QCI class or per period of time. RNI may be used by the mobile edge platform e.g to optimize the mobility procedures ensuring service continuity or inform network-aware and context-aware allocation of resources decision processes.

For connecting MEC Platforms (MEP) in different MEC hosts of the MEC system, ETSI introduced the Mp3 interface in the reference architecture. The Mp3 could be used, for example, for exchange information able to trigger application lifecycle management operations supporting mobility. At the moment, the platform-to-platform interface over Mp3 reference point has not been not specified by ETSI yet.

Even if the first work on MEC steams from the Mobile and virtual machine (VM) ecosystems, the architecture is evolving to embrace a scenario that is basically both access and virtualization technologies agnostic.

Using light non-VM based virtualization technologies such as containers is of primary importance for edge computing deployment in resource constrained environments. As a matter of fact, the recent major report [8] released by the ISG as part of its Phase 2 work studies the impact of alternative virtualization technologies, and specifically it addresses the usage of containers in MEC environments. The results and conclusion of this report highlight that ETSI MEC architectural framework is capable of supporting such technologies, very few updates of existing standards.

Due to its access-agnostic nature, MEC paves the way to 5G guaranteeing a smooth transition from 4G to 5G. The combination of 5G and MEC is a true enabler for Ultra-Reliable Low-Latency communications (URLLC).

In the next session we propose an architecture for Aerial Devices that leverages the ETSI MEC framework.

Novel Proposed Architecture

The proposed system provides an on-demand Multi-Access Edge Computing framework by exploiting the mobility and flexibility of Aerial Platforms. We consider the setup described in Figure 2, where rapid deployment of network connectivity is based on one or more balloons and several drones hovering over the area to provide connectivity and services. Both balloons and drones are equipped with communication interfaces and computing/storage power. In this way, it is possible to define end-to-end slices of the available resources in order to meet the performance requirements and enable the deployment of 5G services. This, in turn, requires proper resource allocation in the two major trunks of the system, i.e. the backhaul connection and the RAN.

In the considered scenario, the backhaul connection is provided by the HAP drone, which has a longer lifetime than mobile BSs, and it can potentially carry bigger weight and cover a relatively large area. The RAN is implemented in a distributed way through a swarm of drones that are positioned in hovering mode in the areas where on-demand coverage is required and can be moved to adapt to the users' distribution.

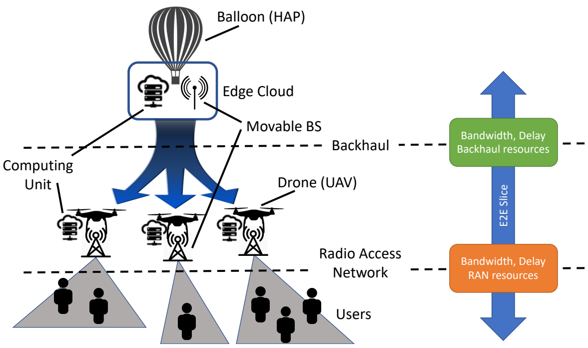

***Figure 2** The proposed architecture and related resources.*

Both the mobile BSs and the HAP drones can be considered as fully operating MEC hosts, deploying ME Applications over a virtualized platform, hosting the ME Platform and exposing services. Information regarding number of users, radio quality, etc. of both UEs (through the drone ME host) and drones (through the balloon ME host) are therefore available through local MEP services (exposed e.g. by the MEC RNI API and MEC Location API) and can be used to better orchestrate the limited resources on the drones. Edge applications running at the mobile BSs or HAP drone edge hosts can send to their respective MEPs their bandwidth requirements (through the Device App API), thus setting the overall system requirements. This architecture requires that edge platform-to-platform communication happens, e.g. through a standardized interface such as the Mp3 interface envisioned by ETSI. Finally, Edge Application orchestration and management can be performed on the balloon or ground data center.

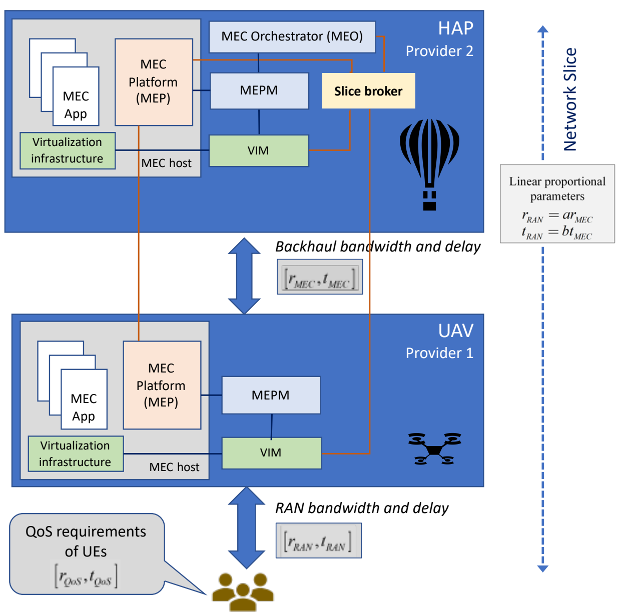

***Figure 3** Logic architecture of the system.*

Figure 3 depicts the logic structure of the proposed system. First, end users are characterized by their requirements according to transmission rate and latency, which are the main components of their required QoS. Second, being Provider 1 the owner of the RAN infrastructure, it dynamically assigns to users network resources, in order to guarantee specific rate and access-network latency. Next, resources are allocated in the backhaul/fronthaul wireless link which connects the mobile BSs (drones) to the aerial MEC data center. This small data center not only hosts computational resources for network tasks but also the network slice manager (Slice Broker) [9]. Moreover, the MEC data center can be positioned on the HAP (e.g. a balloon) or located on the ground and connected to a balloon using an LTE connection.

As a consequence, based on the users' requirements, one or more end-to-end network slices are built, directly connecting the users to the MEC data centers, and providing the requested QoS. Indeed, each network slice is characterized by the allocation of physical resources, mapped into virtual ones, on the RAN and backhaul trunks of the proposed architecture. As a result, a percentage of resources (belonging to the access and the aerial network) will be abstracted and assigned to a group of users in order to satisfy their requirements in terms of QoS.

We assume the following hypotheses for the mobile BSs:

- UAVs are mobile BSs, capable of providing radio connectivity and limited computing processing power (enabling RRH/BBU split and allocation of MEC containers). They also collect information from the end users and transmit it to the aerial platform;
- the mobile BSs can be considered hovering while providing RAN connectivity [10]. This approximation can be considered accurate, since a significant part of the literature is focused on optimal placement of drones to achieve optimal coverage, signal quality and latency on the link drone-end user [11]. Moreover, while such a hypothesis will reduce transmission time and need the deployment of a greater number of drones per time interval, it will positively minimize interference problems among drones and the Doppler effect during transmission (this will also reduce the complexity of the frequency correction).
- The characteristics of the battery and the size of the drones are assumed to carry and supply the hardware of a Pico BS plus a small host for containers (e.g. Raspberry PI platform).

The following assumptions are assigned to the HAP drone and its small data center:

- It can host the UAV orchestration framework in order to reduce the response time for exact placement or small adjustments of position and paths (e.g. similar to the paradigms Follow-Me-Cloud or Follow-Me-Edge) [12]. However, in cases where a nearby MEC data center on the ground is available, the orchestrator might be hosted on the ground.
- The HAP is considered hovering and placed in optimal positions according to the locations of mobile BSs. HAPs

can establish line-of-sight (LoS) backhaul/fronthaul links to the drones they cover. Possible technologies explored in the literature for backhaul are mmWave, LTE and free space optical communications (FSO). The latter is especially efficient and reliable, comparable to a terrestrial backhaul [13]. However, in our scenario, it can be impracticable to equip drones with optical interfaces (small telescopes) to set up the backhaul link. Because of that, we assume a broadband wireless link, which can guarantee up to 120 Mbit/s to end drones requiring a broadband backhaul [13].

- The altitude used by Internet.org consortium for aerial platforms via HAP is about 19 km, in order to provide reliable coverage to a medium-sized urban area. This HAP has an altitude between 18-25km [13]. In the scenario of this article, the aerial data center will have an altitude of about 3 km. In fact, the aerial data center supports backhaul/fronthaul of few mobile BSs (the subsequent simulations will assume a HAP connected to three mobile BSs).

Analysis, Results and Discussion

The scenario and the novel architecture, presented in the previous section, is modeled via a proprietary network simulator written in C++ in order to analyze various aspects and parameters of dynamic network slicing. The RAN is composed by physical (PHY) and data link layer. PHY uses time-division duplexing (TDD) and 20 MHz band.

The analysis of performance and the satisfaction of users' requirements imply the consideration of various trade-offs. The main parameters of these trade-offs are rate, latency, energy consumption and computing capability. The communication is from the MEC HAP drone to the UEs (downlink).

Latency has different components:

- *propagation*, which depends on the distance between network node;
- *transmission*, which is inversely proportional to the available link capacity at a specific time;
- *processing*, which depends on the time needed to complete the computing at the servers.

By increasing the transmission rate, the available capacity decreases thus the transmission delay increases. Regarding the RAN, the analysis takes into account the variability of BS processing time, with LTE transmission downlink delay of 1 ms. Next, augmenting the computing capability of the data centers can significantly affect energy consumption. However, the reduction of the computational speed of the CPUs can reduce energy consumption while increasing processing delay.

By considering a simple model of power consumption of a server given its frequency (which reflects the usage of the CPUs/GPUs) [14], the power consumption is proportional to the cube of the clock frequency of the processing units.

Given a set of computing resources (i.e. CPUs) interconnected by internal data center's links, the Slice Broker assigns a subset of these resources to users in order to get values of processing latency in line with end-to-end latency requirements of users. The number of CPUs at the aerial data center is variable between 10 and 20: such range is acceptable since it is important to consider that the power supply is limited (in fact the aerial platform combines battery and solar power). Each CPU has a frequency of 1 GHz, representing a low-cost energy-efficient solution, acceptable in such scenarios (e.g. a Raspberry pi).

Next, the simulation allows the Slice Broker to assign network slices to three different classes of users: Extreme Mobile Broadband (xMBB) supporting mobile broadband and mobile video streaming, ultra-reliable Machine-Type Communications (uMTCs) (or URLLC) and massive Machine-Type Communications (mMTCs), supporting services such as Internet-of-Things (IoT).

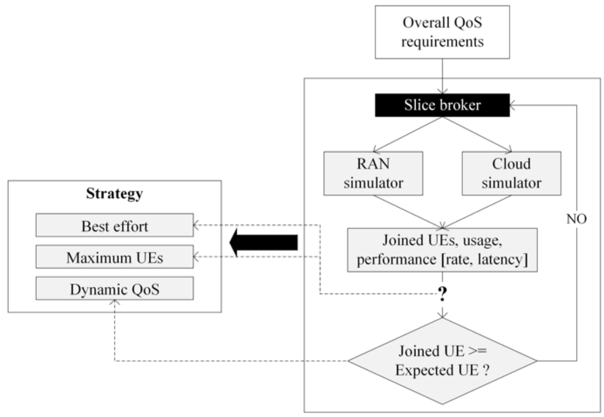

Figure 4 *Flowchart of the Slice Broker's Strategies.*

The Slice Broker [9] considered in this evaluation context can employ different resource management algorithms: deterministic or intelligent. The former is represented by a best-effort-like algorithm, while the latter is realized via reinforcement learning (RL). Figure 4 shows the logic structure of the Slice Broker.

The end user issues an application request to the network, and the broker has to divide the QoS requirements of the user into two parts as a guide for each provider. Based on the QoS requirements, each provider gives the final output, including the system capacity, i.e., the maximum number of users that can be satisfied, the resource usage of the percentages, as well as the true performance per user, such as data rate and latency. It is worth pointing out that the true performance should be no worse than the QoS requirements, otherwise we conclude that these split QoS requirements are not valid. The effective results extend to different resource allocation strategies. The first is the deterministic policy, which either does not need to consider user QoS requirements and directly gives the best performance to every user with the highest data rates and lowest latency, i.e., best-effort-like, either while meeting QoS requirements to allow the maximum number of users to be able to access the service. However, the former strategy has the demerit that it is highly unlikely that the system will be able to serve a sufficient number of users, while the latter policy may lead to a waste of resources, as there will be a situation where many resources are allocated but not really used. Therefore, we propose an intelligent dynamic strategy that considers the expected number of users in advance and then implements a dynamic resource allocation.

Our new intelligent dynamic policy aims to serve all users who expect to join the network, while enhancing the corresponding performance as much as possible, e.g. when there is still a considerable amount of idle resources remaining after meeting the minimum QoS requirements for all anticipated joiners, the smart dynamic policy will allocate idle resources to users, as reflected in an increase in data rate or a decrease in latency. This is inspired by the fact that user traffic is dynamically fluctuating, fortunately, statistics show that traffic has periodic trends, and can be measured and predicted with some degree of accuracy. Benefiting from reinforcement learning [15], by attempting alternative actions and reinforcing tendency actions that produce more rewarding consequences, the optimal strategy can be derived from the interactions with the environment. Thus, the Slice Broker can use RL and a predicted number of users to construct an optimal slicing strategy with respect to each time step.

In RL, one or more states are used to interpret the environment, and in each state, an action is selected based on a certain strategy. Each action leads to a state transition, and the intermediate reward will be used as the numerical evaluation of the selected action. At the end of the training process, the optimal strategy can be derived from the learning experience, and thus RL is widely used for optimization and decision problems.

In order to evaluate our proposed architecture, we have chosen one of the most popular RL techniques, Q-Learning [15], in which the Q-value represents the estimated expectation of the discount cumulative reward for the state-action pair. The values of Q at time steps t and $t+1$ are $Q(S_t, a_t)$, and $Q(S_{t+1}, a_{t+1})$, where a_t and a_{t+1} are the set of available actions for states S_t and S_{t+1}. The maximum Q value implies the best action A_t for state S_t, which derives from $A_t = \arg\max Q(S_t, a_t)$.

In Q-Learning there is a parameter α ($0<\alpha<1$) defined as the learning rate, and it is to balance the knowledge between learned experience and new perception during the training process, while the discount rate γ ($0 \leq \gamma \leq 1$) is to leverage the impact of immediate reward R_t and the potential cumulative rewards received in the future. The Q-value is updated in each training loop, until the terminal state occurs. The training is to end when the episodic accumulative rewards are convergent. Thus, the optimal strategy is derived.

We have referred to multiple users' traffic-flow statistics, and we noticed that it is very common to have the rush hours in the early morning and late afternoon periodically, therefore, combining with the given resource we have designed a simulation scenario to mimic the traffic flow for each hour during the day. In the scenario we assume that

- the system has enough resource to fulfill maximum expected users;
- each user (connected to the same mobile BS) has the same service request, i.e. the same QoS requirement (they belong to the same network slice);
- each user is either static or with very low speed. Particularly, this assumption models verticals such as xMBB, IoT, Industry 4.0, etc. (except for vehicular users);
- the cloud latency only considers the delay received from each node.

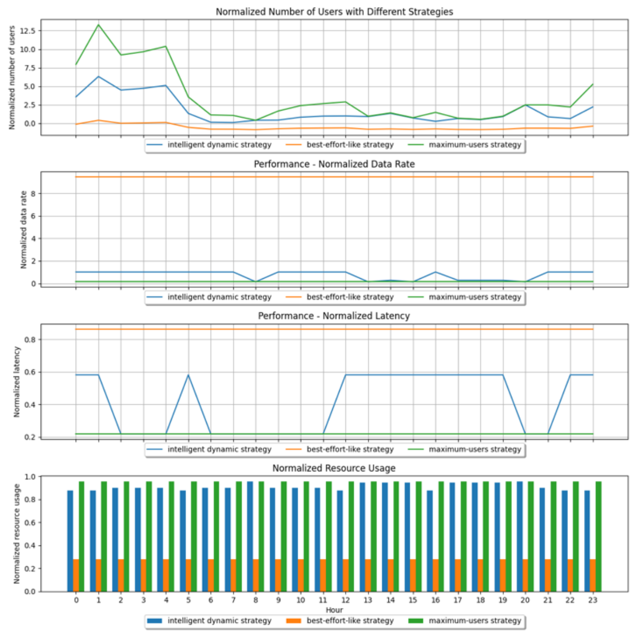

Figure 5 *Performance comparison among three different strategies used by the Slice Broker. Strategy based on reinforcement learning can outperform the deterministic ones by concurrently improving users' performance.*

The Slice Broker splits the user's request regarding data rate and processing latency to both providers, and since in this scenario there are only two providers, we can assume that there are linear relations between the separated QoS requests, as shown in Figure 3. Action is defined as the possible combination of the linear relation parameters, a and b.

The state includes the hourly timestamp and the amount of users. RL is a goal-driven algorithm and in this scenario the construction of reward function is led by the ultimate goal. This means the system should fulfill the QoS requirement for each user and should only serve the expected amount of users instead of the maximum possible. Thus, the system can assign the idle resource to the existing users to improve the performance at the non-rush hours. Therefore, the reward calculation is separately defined under three conditions.

When the QoS cannot be fulfilled with the selected splitting parameters a and b, the reward R is a negative great value, working as a punishment. Once the actual performance has achieved the QoS requirement (if the amount of possible incoming users n_u is higher than the expected amount of users n_e), the reward is then represented as $R=10/(n_u-n_e)$. Since we intend that n_u is only slightly bigger than n_e. Otherwise, it becomes $R=n_u–n_e$, which implies that an opposite situation from the one in the first assumption above. If the system does not have enough resources to fulfill maximum expected users, our reward function guides the intelligent dynamic strategy to act as the maximum-users strategy.

Figure 5 shows the hourly normalized performance of the three strategies. The normalized values are generated using as reference either the amount of expected users or the QoS requirement from different aspects. The orange lines in the plots represent the simulation results using the best-effort-like strategy. From this, we observe that although it provides the highest data rate and the lowest latency, it can only serve a small number of users. Moreover, it is most likely that the best-effort-like strategy cannot allow all the expected users to join the network, as the orange line in the first plot is lower than zero from the fifth hour.

The simulation results derived from the maximum-user policy are plotted as green lines, which show that this policy guarantees all expected users access to network services, but it can only provide each user with the lowest data rate and the maximum latency of the three policies. While the above are static, the blue lines are the performance of our intelligent dynamic algorithm. As it is evident from the first plot, the blue line is dynamically close to zero. While keeping positive, our policy always captures the trend of the expected users per hour. Combining the last three plots, we notice that while serving all the expected users, the normalized usage of our proposed strategy is dynamically balanced. Maximum-users strategy targets users maximization by reserving resources statically, however, in fact often there are fewer service requests, which results in a waste of resource. Unlike maximum-users strategy, our intelligent strategy assigns the remaining available resources to the expected users either reducing latency or increasing the data rate.

Conclusion

UAVs are expected to become key actors in the framework of 5G and B5G networks, especially in scenarios where fast deployment represents a key requirement. This paper presented an architecture to deploy a swarm of drones supported by other HAPs (balloons) in order to bring connectivity and services to users in a given area. The proposal represents an example of solution compatible with current MEC standardization efforts by ETSI and represents a step forward in the integration of mobile aerial platforms within future 5G and B5G systems.

Future work will be aimed at considering HAPs at different altitudes and generalizing the approach to consider different application/QoS scenarios.

References

[1] 3GPP. TS23.501: System Architecture for the 5G System, Dec. 2018.

[2] NGMN Alliance, 5G White Paper V1, 2015. https://www.ngmn.org/work-programme/5g-white-paper.html

[3] 3GPP. TS23.748 Study on enhancement of support for Edge Computing in 5G Core network (5GC).

[4] ETSI GS MEC 003 V2.1.1, Multi-access Edge Computing (MEC); Framework and Reference Architecture. Jan. 2019.

[5] ETSI White Paper No. 28, MEC in 5G networks, ETSI White Paper, Jun. 2018.

[6] ETSI GS MEC 012 V2.1.1, Radio Network Information API, Dec. 2019.

[7] ETSI GS MEC 013 V2.1.1, Location API, Sep. 2019.

[8] ETSI GR MEC 027 V2.1.1, Multi-access Edge Computing (MEC); Study on MEC support for alternative virtualization technologies, Nov. 2019.

[9] K. Samdanis, X. Costa-Perez and V. Sciancalepore, From network sharing to multi-tenancy: The 5G network slice broker, in IEEE Communications Magazine, vol. 54, no. 7, pp. 32-39, July 2016, doi: 10.1109/MCOM.2016.7514161.

[10] R. Bassoli, C. Sacchi, F. Granelli and I. Ashkenazi, A Virtualized Border Control System based on UAVs: Design and Energy Efficiency Considerations, 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1-11, doi: 10.1109/AERO.2019.8742142.

[11] M. Mozaffari, A. Taleb Zadeh Kasgari, W. Saad, M. Bennis and M. Debbah, Beyond 5G With UAVs: Foundations of a 3D Wireless Cellular Network, in IEEE Transactions on Wireless Communications, vol. 18, no. 1, pp. 357-372, Jan. 2019, doi: 10.1109/TWC.2018.2879940.

[12] H. Hellaoui, O. Bekkouche, M. Bagaa and T. Taleb, Aerial Control System for Spectrum Efficiency in UAV-to-Cellular Communications, in IEEE Communications Magazine, vol. 56, no. 10, pp. 108-113, OCTOBER 2018, doi: 10.1109/MCOM.2018.1800078.

[13] Horwath, J., Perlot, N., Knapek, M. and Moll, F., Experimental verification of optical backhaul links for high-altitude platform networks: Atmospheric turbulence and downlink availability. Int. J. Satell. Commun. Network., 25, pp. 501-528, 2007. doi:10.1002/sat.888.

[14] M. Dayarathna, Y. Wen and R. Fan, Data Center Energy Consumption Modeling: A Survey, in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 732-794, Firstquarter 2016, doi: 10.1109/COMST.2015.2481183.

[15] Sutton, Richard S. and Barto, Andrew G., Reinforcement Learning: An Introduction, 2nd Ed., The MIT Press, 2018.

Acknowledgements

This work has been partially funded by NATO Science for Peace and Security (SPS) Programme, in the framework of the project SPS G5428 ”Dynamic Architecture based on UAVs Monitoring for Border Security and Safety”. This work has been partially funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany’s Excellence Strategy – EXC2050/1 – Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden.

Fabrizio Granelli *(fabrizio.granelli@unitn.it) is Associate Professor at the Dept. of Information Engineering and Computer Science (DISI) of the University of Trento (Italy). From 2012 to 2014, he was Italian Master School Coordinator in the framework of the European Institute of Innovation and*

Technology ICT Labs Consortium. He was IEEE ComSoc Distinguished Lecturer for 2012-15, IEEE ComSoc Director for Online Content in 2016-17 and IEEE ComSoc Director for Educational Services in 2018-19. Prof. Granelli is coordinator of the research and didactical activities on computer networks. He is author or co-author of more than 250 papers published in international journals, books and conferences on networking.

Cristina Costa *(ccosta@fbk.eu) is a Researcher staff member of the Fondazione Bruno Kessler, Trento, where she works on Edge Computing, 5G and LoWPAN IoT related topics. Dr Costa has more than twenty years of industrial and academic research experience in various areas of telecommunications and ICT. She has participated in several European collaborative projects and industrial projects, gaining experience both in scientific and management roles. She served as a member of the organising committee of various conferences. She is an IEEE Senior Member and has been appointed secretary of the IEEE Women in Engineering (WIE) Affinity Group of the IEEE Italy Section.*

Jiajing Zhang *(jiajing.zhang@tu-dresden.de) is a Ph.D. student with the Deutsche Telekom Chair of Communication Networks at the Faculty of Electrical and Computer Engineering, Technische Universität Dresden, Germany. She is also member of the DFG Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) and her topic is Intelligent Networks.*

Riccardo Bassoli *(riccardo.bassoli@tu-dresden.de) is senior researcher at the Deutsche Telekom Chair of Communication Networks at the Faculty of Electrical and Computer Engineering, at Technische Universität Dresden, Germany. He received his B.Sc. and M.Sc. degrees in Telecommunications Engineering from University of Modena and Reggio Emilia (Italy) in 2008 and 2010 respectively. Next, he received his Ph.D. degree from 5G Innovation Centre at University of Surrey (UK), in 2016. He was also a Marie Curie ESR at Instituto de Telecomunicações (Portugal) and visiting researcher at Airbus Defence and Space (France). Between 2016 and 2019, he was postdoctoral researcher at University of Trento (Italy).*

Frank H.P. Fitzek *(frank.fitzek@tu-dresden.de) is Professor and Head of the Deutsche Telekom Chair of Communication Networks at Technische Universität Dresden, coordinating the 5G Lab Germany. He is also leading the DFG Cluster of Excellence CeTI. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology – Rheinisch-Westfälische Technische Hochschule (RWTH) – Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002 and became Adjunct Professor at the University of Ferrara, Italy in the same year. In 2003 he joined Aalborg University as Associate Professor and later became Professor.*