AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections. Your article has been peer reviewed, accepted as final, and sent in to IEEE. No text changes have been made to the main part of the article as dictated by the editorial level of service for your publication.

AQ1: Please confirm or add details for any funding or financial support for the research of this article.

AQ2: Please provide the postal code for Dublin City University, Dublin 9, Ireland.

AQ3: Please provide the department name for Reference [31].

An Innovative Machine-Learning-Based Scheduling Solution for Improving Live UHD Video Streaming Quality in Highly Dynamic Network Environments

Ioan-Sorin Comșa, Gabriel-Miro Muntean, *Senior Member, IEEE*, and Ramona Trestian, *Member, IEEE*

***Abstract*—The latest advances in terms of network technologies open up new opportunities for high-end applications, including using the next generation video streaming technologies. As mobile devices become more affordable and powerful, an increasing range of rich media applications could offer a highly realistic and immersive experience to mobile users. However, this comes at the cost of very stringent Quality of Service (QoS) requirements, putting significant pressure on the underlying networks. In order to accommodate these new rich media applications and overcome their associated challenges, this paper proposes an innovative Machine Learning-based scheduling solution which supports increased quality for live omnidirectional (360°) video streaming. The proposed solution is deployed in a highly dynamic Unmanned Aerial Vehicle (UAV)-based environment to support immersive live omnidirectional video streaming to mobile users. The effectiveness of the proposed method is demonstrated through simulations and compared against three state-of-the-art scheduling solutions, such as: static Prioritization (SP), Required Activity Detection Scheduler (RADS) and Frame Level Scheduler (FLS). The results show that the proposed solution outperforms the other schemes involved in terms of PSNR, throughput and packet loss rate.**

***Index Terms*—Omnidirectional video, live streaming, QoS, machine learning, radio resource management, UAV.**

I. Introduction

GLOBAL mobile video traffic continues to grow exponentially, especially with the introduction of Ultra-High-Definition (UHD) or so called 4K video streaming applications. This new application category puts tremendous pressure on the current underlying networks as the average bit rate for 4K video is around 15 to 18Mbps, which is more than double the High Definition (HD) video bit rate and nine times more than the Standard Definition (SD) video bit rate [1].

Additionally, the increasing adoption of new Virtual Reality (VR) and Augmented Reality (AR) enabled high-end mobile devices together with the increasing amount of content ready to be consumed pushes the current 4G networks closer to their saturation. It is expected that the VR/AR generated traffic to continue to follow a high growth trajectory especially with the potential adoption of virtual reality streaming [1] that opens up a new era of 5G-based media services. Moreover, Cisco [1] also predicts that live Internet video will account for 17% of the Internet video traffic by 2022 with IP video traffic reaching 82% of all IP traffic globally.

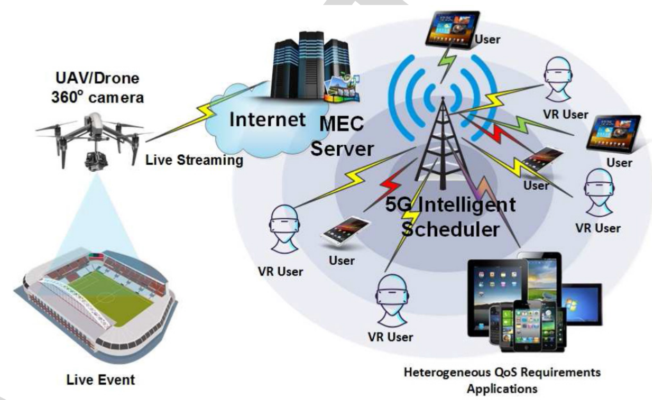


Fig. 1. Highly dynamic immersive live UHD streaming example scenario.

Consequently, in order to keep up with the current and predicted traffic demands, the network operators have already started an accelerated roll-out of 5G communications. As the new 5G technology targets high data rate and very low latency, it opens up a new range of applications starting from immersive augmented reality to driverless cars or even robot-enabled remote surgery. According to Cisco, by 2022, 5G devices and connections will represent more than 3% of global mobile devices and connections, with 12% of the global mobile traffic being generated over the 5G cellular network [1]. However, the network operators need to demonstrate that the tremendous potential of the 5G deployment could meet the users' expectations. The challenge is magnified even further especially given the current wide and diverse range of applications with different Quality of Service (QoS) requirements which need to be supported on a heterogeneity of end-user hardware platforms. Applications such as live network streaming require low latency and jitter, whereas, reliability is needed for applications such as file transfer which cannot

Manuscript received December 6, 2019; revised March 9, 2020; accepted
AQ1 March 12, 2020. *(Corresponding author: Gabriel-Miro Muntean.)*

Ioan-Sorin Comșa is with the Department of Computer Science, Brunel University London, Uxbridge UB8 3PH, U.K. (e-mail: ioan-sorin.comsa@brunel.ac.uk).

AQ2 Gabriel-Miro Muntean is with the School of Electronic Engineering, Dublin City University, Dublin 9, Ireland (e-mail: gabriel.muntean@dcu.ie).

Ramona Trestian is with the Department of Design Engineering and Mathematics, Middlesex University, London NW4 4BT, U.K. (e-mail: r.trestian@mdx.ac.uk).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2020.2983298

tolerate packet loss or high delay. As most applications require end-to-end network support, this hampers the potential development and advantages of new applications. Consequently, it becomes obvious that just increasing the system capacity is not enough to meet the heterogeneous QoS requirements for all mobile users at the same time. This is mainly due to the increasing popularity of bandwidth-hungry applications (e.g., multimedia-based applications), limited radio resources and changeable wireless network conditions. Thus, along with the next generation networks deployment, new emerging technologies and solutions are being explored to help network operators to cope with such high traffic demands, such as: integration of MPEG-DASH [2] as the de-facto video delivery mechanism, Advanced Television Systems Committee (ATSC) 3.0 standard [3], evolved Multimedia Broadcast/Multicast Service (eMBMS) [4], Further eMBMS (FeMBMS) and New Radio MBMS (NR-MBMS) [5], mmWave communications [6], satellite back-haul [7], Software Defined Networks (SDN) and Network Function Virtualisation (NFV) [8], [9], Mobile Edge Computing (MEC) [10], Unmanned Aerial Vehicle (UAV) or drones [11], machine learning [12], etc. As a potential use case of UAV, Mangina *et al.* [13] make use of drones for live streaming for people with limited mobility, so that they could enjoy the immersion as if they were present at the specific location. The aim of this framework is to use the technology to enable opportunities for communication and self expression of people of all levels of physical and cognitive ability.

This work focuses on a highly dynamic mobile scenario involving high bitrate live video streaming, as the one illustrated in Fig. 1. In this scenario, an UAV equipped with an omnidirectional (360°) camera is used to send 4K/8K video captured in real time from a live event taking place for instance in a stadium, to a MEC server attached to a 5G network. VR-enabled users get the live video stream served via the 5G network and expect to enjoy a high quality video experience, as if they were present at the venue. However, to be able to create a high quality immersive experience for the remote users, the network operators need to guarantee low latency and packet loss, and high throughput while also accommodating other traffic classes. Unfortunately, this is not possible to achieve with conventional resource management methods.

In this context, this paper proposes and describes an innovative Machine Learning (ML)-based scheduling solution for radio resource management to improve significantly QoS provisioning and increase users' Quality of Experience (QoE) levels in the presence of heterogeneous traffic. The proposed solution targets particularly highly challenging scenarios which involve live streaming of very high bitrate video in highly dynamic network environments.

The remainder of this article is organized as follows: Section II discusses important related works in this area and Section III presents an overview of the proposed solution. Section IV details the proposed innovative ML-based scheduling solution for increased quality of live high bitrate video streaming in highly dynamic network environments and presents the associated problem formulation. Evaluation results are discussed in Section V in comparison with those of alternative solutions and finally, conclusions are drawn in Section VI.

II. Related Works

A key challenge for network operators is to provide ubiquitous connectivity to different device types and applications with heterogeneous QoS requirements. This challenge is amplified by the increasing popularity of multimedia-based bandwidth-hungry applications with strict QoS requirements that stretch the current 4G networks closer to saturation. Consequently, to be able to accommodate all these new immersive live streaming applications, known for being bandwidth-hungry and having low-latency and packet loss requirements [14], advanced solutions must be adopted to maintain increased QoE for end-users, since QoE is expected to become the biggest differentiator between network operators [15].

An important component that is expected to be integrated within the 5G and beyond 5G networks is the use of UAV [16]. Apart from facilitating temporary radio access and Internet connectivity, UAVs could also be used to facilitate live video broadcasting and enable support for high data rate transmissions [11]. However, to accommodate a high number of users with enhanced QoE levels within the 5G radio access network, system bandwidth needs to be properly managed. According to [17], two adaptation methods classes can be considered to deal with the bandwidth efficiency in order to improve QoS and QoE, such as: passive and active. The active approaches aim to improve the bandwidth allocation by using scheduling algorithms, whereas passive ones refer more to bandwidth-compliant adaptation techniques that adapt the multimedia transmission to the available bandwidth.

As an active adaptation entity, the packet scheduler is responsible for dynamically sharing the system bandwidth between the end-users such that the QoS provisioning is maximized. Different scheduling strategies are proposed in the literature to deal with QoS targets [18]. A scheduler that encapsulates the features of different scheduling strategies is proposed in [19] for 3G downlink systems to assure the multidimensional QoS provisioning under varying traffic and radio channel conditions. However, most of the state-of-the-art schedulers targeting multidimensional QoS requirements aim to prioritize some traffic classes while ignoring others. For instance, Frame Level Scheduler (FLS) [20] prioritizes real-time traffic (e.g., video, voice, gaming) over the more elastic traffic classes (e.g., file transfer, HTTP). In contrast, Required Activity Detection (RADS) [21] prioritizes a group of users according to their packet delay and fairness criterion. However, most of the prioritization schemes are unable to react to the dynamics of the wireless environment, such as: increasing number of users, various traffic characteristics, and changeable network conditions. As a consequence, some traffic classes are over-provisioned while others may have a degraded QoS.

A passive method used for traffic prioritization and bandwidth adaptation is proposed in [17] to manage the transmission of massive clinical applications in high-speed ambulance scenario under variable and limited communication bandwidth.

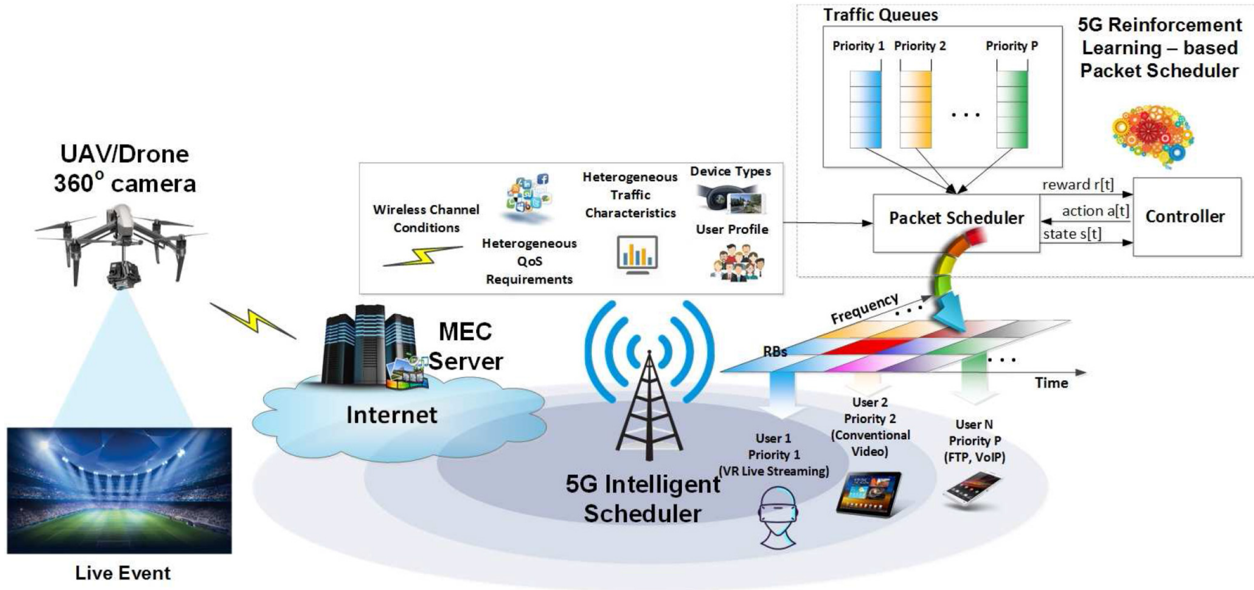


Fig. 2. Proposed 5G UAV-based live streaming framework.

The approach works in two stages: a) the clinical multimedia data is prioritized in four classes based on the disease model and the criticality of each model; b) according to the available bandwidth, different heuristic algorithms are proposed to reduce the clinical data rates according to their priority class. The evaluations show the effectiveness of this approach by transferring the most critical information within the limited bandwidth. By focusing only on QoE improvement, the system bandwidth can remain underutilized. In this sense, a passive adaptation scheme is proposed in [22] to facilitate the video rate adaptation by considering the physical layer information to enable accurate bandwidth estimation. The latest network advancements need to accommodate advanced applications and services with very high data rates and extremely low latency. Wang *et al.* [23] propose the use of fog networking to coordinate a network of drones equipped with cameras to broadcast live events. The objective of the proposed framework is to maximizing the coverage area as well as the available throughput for high-quality video streaming to video servers.

In terms of Radio Resource Management (RRM) and QoS provisioning, classical RRM functionalities would not be able to meet the stringent QoS requirements of all these immersive live streaming applications while also catering for the rest of application classes. In the context of 5G, ML is currently gaining considerable attention as it is seen as one of the key enablers for QoS provisioning [12], [18], [24]–[26] as well as for the development of intelligent services for smart cities [27]. An autonomous network resource management for QoS and QoE provisioning is proposed in [12] to predict the amount of network resources that needs to be allocated to cope with the traffic demands for live and on-demand dynamic adaptive streaming over HTTP. Machine learning is used to optimize the scheduling and resource allocation problems in 5G radio access networks focusing on different combinations of QoS objectives, such as: throughput, delay and packet loss in [18], packet loss and delay in [24], system throughput and user fairness in [25]. However, these ML-based scheduling solutions are designed for homogeneous traffic types only. The ML framework proposed in [26] aims to optimize the resource and power allocation problem for heterogeneous traffic with the scope of improving the delay of Ultra-Reliable and Low-Latency Communications (URLLC) users and throughput of enhanced Mobile Broadband (eMBB) users. Compared to previous works, this paper proposes a ML-based scheduling and resource allocation solution to enable high level of QoS provisioning for mobile users experiencing UAV VR-based live video content while maintaining an acceptable service quality of other traffic types with diverse QoS requirements.

To this extent, the contributions of this paper are two fold:

- an innovative ML-based scheduling solution to enable QoS provisioning for Ultra High Definition video streaming in highly dynamic network environments;
- a QoS-oriented UAV-based integrated system for enabling high quality levels for immersive live video streaming.

The benefits of the proposed ML-based solution compared to other state-of-the-art schedulers are summarized as follows:

- enhanced QoS provisioning (in terms of delay, throughput and packet loss requirements), higher throughput and Peak Signal-to-Noise Ratio (PSNR) for users requesting UHD VR-based live video;
- gains in excess of 100% when monitoring the time fraction when the heterogeneous QoS requirements are met in a mixture of services with various QoS requirements;
- improved inter-class fairness by respecting over time the standard prioritization order; it can accommodate a higher number of UHD VR video connections and avoids the over/under-provisioning of other traffic classes.

III. Proposed Framework for UAV-Based 4K Streaming

The main components of the proposed quality and performance-oriented system for high quality live video streaming are illustrated in Fig. 2. The figure presents a very challenging deployment involving a UAV with a 360° camera, a MEC server, a 5G intelligent packet scheduler and VR users. The UAV has a 360° spherical camera that records a live event (e.g., football games, concerts, festivals, etc.). The UAV communicates via the 5G network on the ground to send 4K/8K UHD video to the MEC server. For simplicity, it is assumed that there is no loss on the communication link between the UAV and the MEC server. The MEC server will then stream live the UHD video content to the users. However, in order to accommodate a heterogeneous traffic mix with different QoS requirements, an intelligent ML-based packet scheduler is proposed to enable high QoS provisioning for different traffic classes, including for live high bitrate video streaming. The mix of traffic can consider the 5G services and use cases such as eMBB, URLLC and massive Machine Type Communications (mMTC) as well as other types of 4G related services with more relaxed QoS requirements.

The role of the packet scheduler is to allocate the available frequency resources to active users within a given cell to improve as much as possible the fraction of scheduling time when the QoS requirements are met for each traffic type. The scheduling process is conducted at each Transmission Time Interval (TTI) and usually works in two steps: a) Time-based Prioritization (TP) where a group of users with more stringent QoS requirements is prioritized among other users with more relaxed QoS constraints and b) Frequency-based Prioritization (FP) that aims to allocate the radio resources in order to increase the QoS provisioning in terms of delay, packet loss and rate requirements for the pre-selected group of users. While time prioritization is seen as an outer QoS provisioning scheme for all traffic classes based on a given priority order, frequency prioritization acts as an inner QoS provisioning scheme for the pre-selected users. Consequently, the scheduler will prioritize data packets in both time and frequency domains based on current networking conditions that may change at each TTI, including: number of users for each traffic class, QoS profiles, heterogeneous QoS parameters, VR live streaming characteristics, channel conditions, etc. However, many existing scheduling schemes are not able to adapt to the dynamic and unpredictable networking conditions [18]. For instance, some time-based prioritization schemes aim to over-provision some traffic classes while degrading the performance of others [20], [21], whereas the frequency-based prioritization techniques will address only particular QoS requirements at any time [18]. In order to avoid these drawbacks, the proposed scheduling solution is flexible, being able to adapt according to the current network conditions in order to enhance the fraction of time when the heterogeneous QoS requirements are respected.

Since live UHD VR-based video streaming has strict QoS requirements with data rates at least twenty times greater than other conventional applications [1], the best practice would be to decide at each TTI the most suitable traffic class to be prioritized in order to: a) meet the very stringent QoS requirements of live UHD VR-based traffic and b) avoid the starvation effect for other types of applications. In the frequency domain, the most suitable scheduling rule is selected to improve the QoS provisioning for each selected traffic class. Therefore, an intelligent ML-based solution is introduced to learn over time and propose the most suitable prioritization decisions based on current scheduler states. Therefore, this paper proposes an innovative ML-based scheduler for heterogeneous traffic in Orthogonal Frequency Division Multiple Access (OFDMA) downlink systems. The proposed ML-based scheduling solution is able to take each time two scheduling decisions in order to increase the amount of time when all QoS requirements are met. This two-dimensional decision prioritizes a certain traffic class at each TTI and decides the scheduling rule that allocates the available bandwidth to users of the pre-selected class in the frequency domain.

IV. Intelligent ML-Based Scheduling Solution

As previously stated, the proposed ML-based scheduler (see Fig. 2) is able to select at each TTI the most suitable traffic class to be prioritized in time domain and the best scheduling rule for the user prioritization in frequency domain in order to improve the QoS provisioning. These decisions could be taken based on various parameters, such as: wireless channel conditions, application requirements, traffic characteristics, users' profile, device types, etc. The details of the ML-based scheduler are presented next in this section.

A. Prioritization-Based Scheduling

In frequency domain, it is considered that the available bandwidth is divided in equal Resource Blocks (RBs), the smallest radio resource that can be allocated by the Base Station (BS) to the user (see Fig. 2). We define by $\mathcal{B} = \{1, 2, \dots, B\}$ the set of available RBs in a given bandwidth. To get the necessary bandwidth needed to accommodate a high number of UHD VR-enabled live video streaming connections, we aggregate multiple radio bandwidths. Each User Equipment (UE) is characterized by a single traffic class, with a given priority and a QoS profile in terms of delay, packet loss and throughput requirements. Multiple UEs may request different services with heterogeneous QoS requirements. A successful scheduler should be able to accommodate UHD VR-based live services as well as other conventional traffic types (e.g., video, voice, file transfer, etc) without penalizing one over the other. The list of symbols used in this paper is presented in Table I.

Let us consider P the number of traffic classes with different QoS profiles. We define by $\mathcal{P} = \{1, 2, \dots, P\}$ the priority set such that traffic class 1 has the highest priority (i.e., UHD VR-based live streaming traffic) while traffic class P has the lowest priority. The *Static prioritization (SP)* is defined according to the 3GPP guidelines [28] as follows: regardless of the network conditions, the scheduling process respects the priority set $\mathcal{P} = \{1, 2, \dots, P\}$ for the entire downlink transmission session. Let us define the set of active users for all classes as $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_P\}$, where $\mathcal{U}_p$ is the subset of users corresponding to traffic class $p \in \mathcal{P}$. We denote by U_p the number of users belonging to class $p \in \mathcal{P}$, while by U, the total number of active users from all classes. Moreover, the set of heterogeneous QoS objectives in terms of their requirements' accomplishment is defined as $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_P\}$,

TABLE I
LIST OF NOTATIONS

Parameter	Description
$\mathcal{A}$	Discrete and two-dimensional controller action space
$\mathbf{a}[t]$	Current action $\mathbf{a} \in \mathcal{A}$ decided at TTI t
$\mathcal{B}$	Set of resource blocks from different carriers
b	Random resource block $b \in \mathcal{B}$
B	Max. no. of resource blocks
E_c	Error of critic neural network
E_a	Error of actor neural network
L_H	Number of hidden layers
$m_{b,p,u}$	Metric of user $u \in \mathcal{U}_p$ on RB $b \in \mathcal{B}$
N_l	Number of nodes corresponding to layer l
$\mathcal{O}$	Set of heterogeneous objectives
$\mathcal{O}_p$	Set of objectives corresponding to class p
o	Objective index belonging to a given set $\mathcal{O}_p$
O_p	Number of QoS objectives for the traffic class $p \in \mathcal{P}$
$\mathcal{P}$	Set of traffic classes in the priority order given by [28]
p	Random traffic class $p \in \mathcal{P}$
P	Max. no. of traffic classes
$\mathcal{R}$	Set of scheduling rules
r	Random scheduling rule $r \in \mathcal{R}$
R	Max. no. of scheduling rules from $\mathcal{R}$
$\mathcal{S}$	Continuous and multi-dimensional scheduler state space
$\mathbf{s}[t]$	Current scheduler state $\mathbf{s} \in \mathcal{S}$ at TTI t
$\mathcal{U}$	Set of heterogeneous users
$\mathcal{U}_p$	Set of users corresponding to class p
u	User index belonging to a given class $\mathcal{U}_p$
U_p	Number of active users from $\mathcal{U}_p$
U	Total number of heterogeneous users
$x_{o,p,u}$	QoS indicator of $o \in \mathcal{O}$ and user $u \in \mathcal{U}_p$
$\bar{x}_{o,p,u}$	QoS requirement of $o \in \mathcal{O}$ and user $u \in \mathcal{U}_p$
$\Gamma_{d,u}$	Utility function of rule $r \in \mathcal{R}$ and user $u \in \mathcal{U}_p$
$\rho[t+1]$	System reward value received at TTI t+1

where $\mathcal{O}_p$ is the set of objectives for class $p \in \mathcal{P}$. It is said that set $\mathcal{O}_p$ is met if the delay, packet loss and throughput requirements are respected by all active users belonging to traffic class $p \in \mathcal{P}$.

In frequency domain, the process of user scheduling and resource allocation is conducted according to a given scheduling rule that is oriented on a particular QoS objective or on a group of QoS objectives. We define the set of scheduling rules as $\mathcal{R} = \{1, 2, \ldots, R\}$, where R represents the maximum number of rules. Assuming that a SP scheme is employed at this stage at each TTI, the set of active users $\mathcal{U}_1$ is passed in the frequency domain for scheduling. Here, a given scheduling rule $r \in \mathcal{R}$ contributes to the metric computation for each user $u \in \mathcal{U}_1$ on each RB $b \in \mathcal{B}$. Each metric shows how necessary is for each user $u \in \mathcal{U}_1$ to get each resource $b \in \mathcal{B}$ from the perspective of the addressed objective $o \in \mathcal{O}_1$ targeted by the scheduling rule $r \in \mathcal{R}$. In the initial phase of scheduling, a number of U_1 metrics is computed for each RB $b \in \mathcal{B}$ by summing a total number of $U_1 \cdot B$ metrics. In the second phase, the scheduler allocates each RB $b \in \mathcal{B}$ to the user with the highest metric and the process is repeated RB-by-RB until the entire set $\mathcal{B}$ is allocated. However, some metrics can be zero since the QoS objectives are met or there are not enough packets in the queue for some users. If all metrics are equal, then the RB $b \in \mathcal{B}$ remains unoccupied. Finally, the third phase of the scheduling process aims at calculating the size of the transport block for each user scheduled on different RBs and determines the modulation and coding scheme necessary to decode the data at the reception. The scheduling process can be repeated for the next prioritized class (i.e., $p = 2$) if some RBs are unoccupied once the users from $\mathcal{U}_1$ are scheduled.

By employing this SP scheme, the UHD VR-based live video streaming traffic is always allocated the best resources while adversely affecting QoS provisioning for other traffic classes. To avoid this fundamental drawback, other traffic classes must be prioritized when network conditions are favorable. Consequently, in this work, the proposed approach aims to select at each TTI the traffic class $p \in \mathcal{P}$ in such a way that the satisfaction of heterogeneous QoS requirements has the highest possible outcome under the current networking conditions. In this way, we decide at each TTI the prioritization set $\mathcal{P}[t] = \{p, 1, \ldots, p-1, p+1, \ldots, P\}$, where class $p \in \mathcal{P}$ gets as many resources as needed up to the maximum number of RBs, whereas other classes receive the remaining resources by following the priority order of $\{1, \ldots, p-1, p+1, \ldots, P\}$. Even so, if always applying the same scheduling rule for frequency prioritization, only one objective across all traffic classes would be addressed, while harming the performance of other QoS targets. Consequently, in the frequency domain, our aim is to apply at each TTI the most suitable scheduling rule in order to increase the fraction of time (in TTIs) when the heterogeneous QoS requirements are met.

B. Multi-Class and Multi-Objective Optimization Problem

Let us define by $x_{p,u,o}$ the Key Performance Indicator (KPI) of user $u \in \mathcal{U}_p$ and objective $o \in \mathcal{O}_p$ and by $\bar{x}_{p,u,o}$ its associated requirement. It is said that user $u \in \mathcal{U}_p$ meets objective $o \in \mathcal{O}_p$ if and only if $x_{p,u,o}$ respects $\bar{x}_{p,u,o}$. Furthermore, let us define the current KPI vector $\mathbf{x}_{p,u}[t] = [x_{p,u,o_1}, x_{p,u,o_2}, \ldots, x_{p,u,O_p}]$ and its associated requirement vector $\bar{\mathbf{x}}_{p,u} = [\bar{x}_{p,u,o_1}, \bar{x}_{p,u,o_2}, \ldots, \bar{x}_{p,u,O_p}]$. User $u \in \mathcal{U}_p$ meets all QoS objectives if and only if $\mathbf{x}_{p,u}$ respects the requirement vector $\bar{\mathbf{x}}_{p,u}$. By extending this reasoning, the entire set of objectives is met for each traffic class $p \in \mathcal{P}$, if vector $\mathbf{x}_p[t] = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \ldots, \mathbf{x}_{p,U_p}]$ respects its requirements $\bar{\mathbf{x}}_p = [\bar{\mathbf{x}}_{p,1}, \bar{\mathbf{x}}_{p,2}, \ldots, \bar{\mathbf{x}}_{p,U_p}]$. The proposed framework aims to increase the number of TTIs when the KPI vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P]$ respects the QoS requirement vector $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_P]$. We formulate in (1) the multi-class and multi-objective optimization problem that aims to determine at each TTI the most convenient traffic class to be prioritized and scheduling rule to be applied in the frequency domain such that vector of QoS indicators $\mathbf{x}$ reaches the highest possible outcome when reporting to the vector of QoS requirements $\bar{\mathbf{x}}$.

$$\begin{aligned} \max_{i,j,k} \; & \sum_{r\in\mathcal{R}} \sum_{p\in\mathcal{P}} \sum_{u\in\mathcal{U}_p} \sum_{b\in\mathcal{B}} i_{r,p}[t] \cdot j_{p,u}[t] \cdot k_{u,b}[t] \cdot \Gamma_{r,p}(\mathbf{x}_{p,u}[t]) \\ & \times \gamma_{u,b}[t], \qquad (1) \\ s.t. \; & \sum_{u} k_{u,b}[t] \leq 1, \quad b = 1, \ldots, B, \qquad (1a) \\ & \sum_{p} j_{p,u}[t] \leq 1, \quad u = u_1, \ldots, u_{U_p}, p = 1, \ldots, P, \qquad (1b) \\ & \sum_{u} j_{p^*,u}[t] = U_{p^*}, \quad p^* \in \mathcal{P}, \qquad (1c) \\ & \sum_{u} j_{p^\otimes,u}[t] = 0, \quad \forall p^\otimes \in \mathcal{P}\backslash\{p^*\}, \qquad (1d) \\ & \sum_{r} i_{r,p}[t] = 1, \quad p = 1, 2, \ldots, P, \qquad (1e) \end{aligned}$$

$$\sum_{p} i_{r^*,p}[t] = P, \quad r^* \in \mathcal{R}, \tag{1f}$$

$$\sum_{p} i_{r^{\otimes},p}[t] = 0, \quad \forall r^{\otimes} \in \mathcal{R} \backslash \{r^*\}, \tag{1g}$$

$$i_{r,p}[t] \in \{0, 1\}, \quad \forall r \in \mathcal{R}, \forall p \in \mathcal{P}, \tag{1h}$$

$$j_{p,u}[t] \in \{0, 1\}, \quad \forall p \in \mathcal{P}, \forall u \in \mathcal{U}_p, \tag{1i}$$

$$k_{u,b}[t] \in \{0, 1\}, \quad \forall u \in \mathcal{U}_p, \forall b \in \mathcal{B}. \tag{1j}$$

In (1) $\gamma_{u,b}[t]$ is the achievable user rate that quantifies the number of bits transmitted if the RB $b \in \mathcal{B}$ would be allocated to user $u \in \mathcal{U}_p$. Basically, $\gamma_{u,b}[t]$ is determined based on the Channel Quality Indicator (CQI), a bandwidth dependent vector reported by each user $u \in \mathcal{U}_p$ to the base station. For each scheduling rule $r \in \mathcal{R}$, a unique utility function $\Gamma_{r,p}(\mathbf{x}_{p,u})$ is associated in order to attenuate the channel variations given by $\gamma_{u,b}[t]$ and to provide to the user the priority to be scheduled in the frequency domain. Any utility function $\Gamma_{r,p}(\mathbf{x}_{p,u}) : \mathbf{R} \rightarrow \mathbf{R}$ must be monotone and concave [29]. The utility functions can be designed in many ways by considering different KPIs as arguments with certain impact when meeting the heterogeneous and multidimensional QoS requirements. More examples of utility functions are presented in the next section. When setting the same utility function $\Gamma_{r,p}(\mathbf{x}_{p,u})$ for all traffic classes, no matter what the prioritization set $\mathcal{P}_p[t]$ is, the KPI vector $\mathbf{x}$ respects the requirement vector $\bar{\mathbf{x}}$ in a certain measure. The idea is to select at each TTI the prioritization set $\mathcal{P}_p[t]$ and the most suitable utility such that the QoS provisioning would be maximized.

The traffic class, scheduling rule and radio resources are assigned based on the decision variables. In (1), $k_{u,b}[t]$ is the resource allocation variable: $k_{u,b}[t] = 1$ when RB $b \in \mathcal{B}$ is allocated to UE $u \in \mathcal{U}_p$ and $k_{u,b}[t] = 0$, otherwise. Constraints in (1a) aim to allocate at most one user to each RB. Variable $j_{p,u}[t]$ assigns each user to a specific traffic class. Constraints (1b) indicate that each user belongs to at most one traffic class. Constraints (1c) and (1d) show that only users from the selected traffic class $p^* \in \mathcal{P}$ are passed in the frequency domain. Variable $i_{r,p}[t]$ determines the type of utility to be selected at each TTI. Constraints (1e) indicate that one type of utility function per traffic class is selected at each TTI, whereas constraints (1f) and (1g) show that the same scheduling rule is selected for all traffic classes, where variable $r^* \in \mathcal{R}$ is the selected scheduling rule at TTI t and $r^{\otimes} \in \mathcal{R}$ are the other scheduling rules remained un-selected at TTI t. Constraints (1h), (1i) and (1j) make the entire problem combinatorial.

Due to very high complexity, solving the optimization problem from (1) at each TTI is difficult to achieve. Thus, we propose a sub-optimal solution aiming to split this problem in two sub-problems: in the first sub-problem, the prioritization set $\mathcal{P}_p[t]$ is decided and the most appropriated scheduling rule $r \in \mathcal{R}$ is assigned; in the second sub-problem, the resource allocation is performed based on the prioritized traffic class and selected scheduling rule. For the first sub-problem, we propose a ML-based approach [30] to decide at each TTI the class $p^* \in \mathcal{P}$ to be prioritized at first and the best fitting scheduling rule $r^* \in \mathcal{R}$ for the resource allocation. The second sub-problem aims to solve the user scheduling from $\mathcal{U}_{p^*}$ and the resource allocation based on the selected scheduling rule $r^* \in \mathcal{R}$ as described in Section IV-A. As a first step of the scheduling process, we determine the metric $m_{b,p^*,u}$ for each user $u \in \mathcal{U}_{p^*}$ and RB $b \in \mathcal{B}$ at each TTI as follows:

$$m_{b,p^*,u}[t] = \Gamma_{r^*,p^*}\big(\mathbf{x}_{p^*,u}\big) \cdot \gamma_{u,b}[t]. \tag{2}$$

As a result, the matrix of metrics $\mathbf{m} = [m_{b,p^*,u}] \in \mathbb{R}^{U_{p^*} \times B}$ is computed, where $b = \{1, 2, \ldots, B\}$ and $u = \{u_1, u_2, \ldots, u_{U_{p^*}}\}$. For each RB $b \in \mathcal{B}$, a vector of metrics is considered, such as: $\mathbf{m}_b = [m_{b,p^*,u_1}, m_{b,p^*,u_2}, \ldots, m_{b,p^*,u_{U_{p^*}}}]$. Resource $b \in \mathcal{B}$ is allocated to that user that has the maximum metric value from the vector $\mathbf{m}_b$, written in the following manner:

$$b \mapsto u, \quad if \quad u = argmax_{u'}\big(m_{b,p^*,u'}[t]\big), \tag{3}$$

where expression $b \mapsto u$ allocates RB b to user u and $k_{u,b} = 1$. It is important to mention that the allocation is performed RB-by-RB until the entire set of RBs $\mathcal{B}$ gets allocated. However, if for example $\mathbf{m}_{b'} = [0, 0, \ldots, 0]$, then RB $b' \in \mathcal{B}$ remains unoccupied. This resource can be allocated when the scheduling process is repeated for the next prioritized traffic class from the remained set of $\mathcal{P}[t] \backslash \{p^*\}$. By following this model, under certain network conditions it might happen that not all the users could get enough resources to meet their QoS objectives. The aim of the proposed scheduler is to increase as much as possible the QoS provisioning for UHD VR video users with insignificant QoS degradation of other services by properly selecting each time the traffic class to be prioritized and the scheduling rule to be performed in the frequency domain.

C. Types of Scheduling Rules

A scheduling rule $r \in \mathcal{R}$ provides a unique utility function $\Gamma_{r,p}(\mathbf{x}_{p,u})$ focused on a particular or a group of QoS objectives. User fairness is one of the most popular objectives which can be addressed when employing the following function [31]:

$$\Gamma_{1,p}\big(\bar{T}_{p,u}\big) = 1/\bar{T}_{p,u} \tag{4}$$

where $\bar{T}_{p,u}$ is the average throughput of user $u \in \mathcal{U}_p$ calculated based on the exponential moving filter and the scheduling rule $r = 1$ is Proportional Fair (PF). According to (2), (3) and (4), user $u \in \mathcal{U}_p$ with the highest ratio between achievable rate and average throughput on RB $b \in \mathcal{B}$ is selected, while keeping a certain fairness with the previously served users.

Guaranteeing the Bit Rates (GBR) is another QoS objective that can be addressed when selecting the function [32]:

$$\Gamma_{2,p}\Big(\bar{\bar{T}}_{p,u}\Big) = [1 + w_1 \cdot e^{-w_2 \cdot \left(\bar{\bar{T}}_{p,u} - T^R_{p,u}\right)}] \cdot \Gamma_{1,p}\big(\bar{T}_{p,u}\big). \tag{5}$$

where $\bar{\bar{T}}_{p,u}$ is the average user throughput calculated with the median moving filter and $r = 2$ is the Barrier Function (BF) scheduling rule. Users with lower average rates than that of the corresponding requirements $T^R_{p,u}$ are preferred to be scheduled on each RB.

Delay objective aims at respecting the Head-of-Line (HoL) packet delay of each user at each TTI. One possible solution to achieve this target is to employ the following function [33]:

$$\Gamma_{3,p}\big(D_{p,u}\big) = e^{w_3 \cdot D_{p,u}/D^R_{p,u}} \cdot \Gamma_{1,p}\big(\bar{T}_{p,u}\big), \tag{6}$$

where $D_{p,u}$ is the HOL delay of user $u \in \mathcal{U}_p$ at TTI t, $D^R_{p,u}$ is the corresponding requirement and $r = 3$ is entitled the EXPonential (EXP) rule. Users with packets approaching to their deadline receive a much higher priority to be scheduled given the exponential function.

The Packet Loss Rate (PLR) of each user can be improved when the scheduler employs the following utility function [34]:

$$\Gamma_{4,p}\big(L_{p,u}\big) = w_4 \cdot L_{p,u}/L^R_{p,u} \cdot \Gamma_{1,p}\big(\bar{T}_{p,u}\big), \tag{7}$$

where $L_{p,u}$ is the PLR value at TTI t of user $u \in \mathcal{U}_p$, $L^R_{p,u}$ is the corresponding PLR requirement and $r = 4$ is the Opportunistic Packet Loss Fair (OPLF) scheduling rule. When the throughput, delay and PLR requirements are met by all users, BF, EXP and OPLF, respectively act similar to the PF scheduling rule.

D. Controller and Packet Scheduler Interaction

In order to increase the fraction of scheduling time when the heterogeneous QoS requirements are respected, we propose the use of Reinforcement Learning (RL) [30] to learn the most suitable traffic prioritization and scheduling rule that can be applied in real time scheduling. RL makes use of an agent (e.g., intelligent controller) that in time will learn to take actions which will generate the maximum reward by interacting with the environment (e.g., packet scheduler). As seen from Fig. 2, at TTI t, the controller observes a state $\mathbf{s}[t] \in \mathcal{S}$, representing the current network conditions, and takes an action $\mathbf{a}[t] = [p, r] \in \mathcal{A}$ that prioritizes traffic class $p \in \mathcal{P}$ in time domain and selects the scheduling rule $r \in \mathcal{R}$ to be applied in the frequency domain. The scheduling procedure is conducted based on the selected action and the system evolves to the next state $\mathbf{s}[t+1] = \mathbf{s}' \in \mathcal{S}$ at TTI $t+1$. As illustrated in Fig. 2, the reward value received from the scheduling environment evaluates the performance of the applied action in the previous state. This function is calculated based on the set of KPIs $\mathbf{x}[t+1] = \mathbf{x}'$ received at TTI $t+1$. If we define the reward function as $\rho : \mathcal{X} \to [-1, 1]$, where $\mathcal{X} \subset \mathcal{S}$ is the state space of KPI vectors, then the proposed function takes the following form:

$$\rho\big(\mathbf{s}'\big) = \sum_p \sum_o w_p \cdot \rho_{p,o}\Big(\mathbf{x}'_p\Big), \tag{8}$$

where $\rho_{p,o}$ is the reward value of traffic class $p \in \mathcal{P}$ and objective $o \in \mathcal{O}_p$, respectively. In (8), $\mathbf{x}'_p$ is the KPI vector of class $p \in \mathcal{P}$ at TTI t+1. This $\rho_{p,o}$ value denotes how far the online KPI parameters of traffic class $p \in \mathcal{P}$ are from their requirements in terms of objective $o \in \mathcal{O}_p$. The weight w_p sets the 3GPP priority for each class as denoted by the static prioritisation set $\mathcal{P}$. The controller must explore a high number of state-to-state transitions to optimize the prioritization decisions.

E. RL-Based Scheduling Framework

Since the scheduler state space is multi-dimensional and continuous, the scheduling problems cannot be enumerated exhaustively. We can only approximate the best traffic class to be prioritized and the scheduling rule to be performed in the frequency domain, such that the QoS provisioning is much improved. To reduce the complexity for the learning framework, Neural Network (NN) is used to approximate the best prioritization decisions at each current state. During the learning stage, the NN weights are updated at each TTI based on the scheduler and controller interaction as shown in Fig. 2. In the exploitation stage, these weights are saved and the neural network is implemented as a non-linear function.

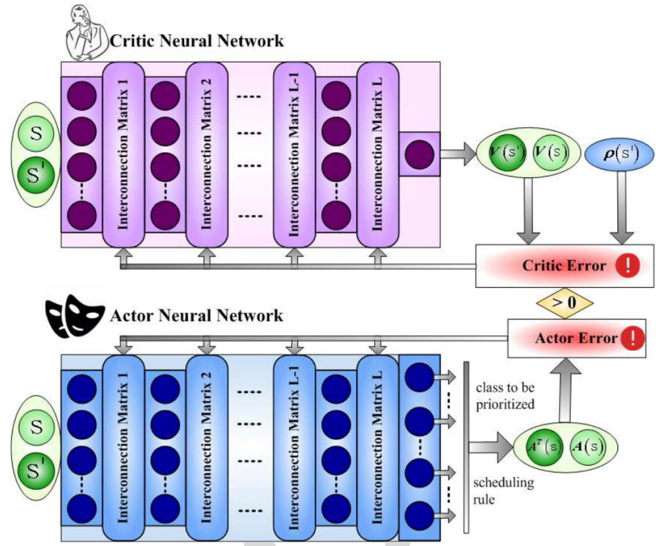


Fig. 3. CACLA-based RL controller architecture.

We propose the implementation of RL framework with a minimum complexity. In this sense, let M be the number of NN output pins in which, the first $M/2$ pins can be used to determine the index of the traffic class to be prioritized and the rest of output pins to decide the scheduling rule to be applied in the frequency domain. To train this non-linear function with multi-dimensional input and output variables, we use Continuous Actor-Critic Learning Automata (CACLA) algorithm [35]. As seen from Fig. 3, CACLA considers two neural networks: a) the critic neural network that approximates the state value function and criticizes the action taken on each state; b) actor neural network that approximates the best prioritization set $\mathcal{P}_p[t]$ and scheduling rule $r \in \mathcal{R}$ to be applied on each state. The role of the critic function is to examine the actor activity and improve its decisions over time.

As an internal structure, a neural network is composed by L number of layers, including here the hidden and output layers only. Therefore, we define the number of hidden layers as $L_H = L - 1$. Each layer $l \in \{1, 2, \ldots, L+1\}$ is composed by neurons or nodes and interconnection matrices that represent the weights connecting the nodes within two consecutive layers, for example l and $l+1$. If N_l and N_{l+1} are the number of nodes (not including the bias nodes) of layers l and $l+1$, respectively, then the total number of weights to be updated at each TTI is $\sum_{l=1}^{L} (N_l + 1) \cdot N_{l+1}$. As indicated in Fig. 3, when CACLA algorithm is employed, two sets of weights need to be updated since both actor and critic neural networks are involved during the learning stage.

The functional structure of critic NN is taking the form of the non-linear function defined as: $V : \mathcal{S} \rightarrow [-1, 1]$. The actor NN takes the same form with the amendment that the output value is multi-dimensional and the definition domain is $A : \mathcal{S} \rightarrow [-1, 1]^M$. In the learning stage, two steps are performed at each TTI: a) the updating step in which the weights of both neural networks V and A are updated according to CACLA algorithm and b), the action selection step, that determines the policy of how the controller action is selected at each TTI. In the exploitation stage, only the learnt actor function is used to provide the M dimensional decision under the form of the controller action $\mathbf{a}[t+1] = [p, r]$ that can be decoded into traffic class prioritization and scheduling rule selection.

The updating process based on CACLA algorithm aims to refine the weights of both networks iteratively, on each state. For example, when the current state is $\mathbf{s}' \in \mathcal{S}$, the error between the impact of applied action $\mathbf{a}[t] \in \mathcal{A}$ in the previous state $\mathbf{s}[t] \in \mathcal{S}$ and its expectation must be reinforced through the neural networks. Since CACLA makes use of two neural networks, then two types of errors must be reinforced.

Critic Error: At the beginning of the learning stage, the weights of the critic NN are randomly chosen. Thus, these weights are gradually updated based on the quality of the applied actions in every state. As seen in Fig. 3, the adaptation of the critic NN weights comprises two steps: a) *forward propagation* responsible to get the consecutive critic values $\{V(\mathbf{s}), V(\mathbf{s}')\} \in [-1, 1]$ in order to quantify the impact of action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{s} \in \mathcal{S}$; and b) *back-propagation* step that calculates the critic error and propagates it through the critic NN based on the gradient descent principle [35]. Without going into details, the gradient descent calculates the error for each neuron of each layer $l \in \{2, \ldots, L+1\}$ and updates the weights accordingly. The critic error function $E_c : \mathcal{S} \times \mathcal{S} \rightarrow [-1, 1]$ is defined (9), where $\{V^T(\mathbf{s}), V(\mathbf{s})\}$ are determined by propagating the states $(\mathbf{s}, \mathbf{s}')$ through the critic NN from input to the output layers:

$$E_c(\mathbf{s}', \mathbf{s}) = V^T(\mathbf{s}) - V(\mathbf{s}). \tag{9}$$

Here, the target value is determined as $V^T(\mathbf{s}) = \rho + \gamma \cdot V(\mathbf{s}')$, where $\gamma \in [0, 1]$ is a discount factor and ρ is the reward value calculated with (8).

Actor Error: If the critic error is positive $E_c(\mathbf{s}', \mathbf{s}) \geq 0$, then the previous action was a good choice and the actor NN can be updated as well. If $E_c(\mathbf{s}', \mathbf{s}) < 0$, then the previous action was an unfortunate choice and then, the actor NN must be discouraged in taking such decision in the future. Consequently, the actor NN is not updated. When $E_c(\mathbf{s}', \mathbf{s}) \geq 0$, the actor NN is updated by following the same forward and backward propagation principles. The multi-dimensional actor error is determined based on the function $E_a : \mathcal{S} \rightarrow [-1, 1]^M$:

$$E_a(\mathbf{s}) = A^T(\mathbf{s}) - A(\mathbf{s}), \tag{10}$$

where A^T is the target multi-dimensional action value determined based on some probability distributions. At the beginning of the learning stage, it is not recommended to exploit the actor NN decisions and then, a random multi-dimensional value of $A^T(\mathbf{s})$ different from $A(\mathbf{s})$ is preferred in order to enlarge the exploration of the scheduler state space. This is denoted as the *improvement* step. Once the learning process is approaching to its deadline, we aim to exploit more the actor decisions and then, the multi-dimensional target $A^T(\mathbf{s})$ is equal to $A(\mathbf{s})$. This is denoted as the *exploitation* step. For an optimal learning, it is preferred to mix improvement and exploitation steps with certain probabilities. Certainly, more improvements steps are preferred at the beginning of the learning stage, whereas the end of the learning stage is likely to use more exploitation steps. In this way, we monitor if the mean actor error can converge or not to certain error levels. Once the neural network(s) is(are) updated, the RL controller decides the new action $\mathbf{a}' \in \mathcal{A}$ to be applied in state $\mathbf{s}' \in \mathcal{S}$.

V. System Evaluation

The proposed adaptation framework was implemented in the RRM Scheduler Simulator [31], which is a C/C++ object oriented tool that inherits the LTE-Sim simulator [36]. For the performance evaluation, an infrastructure of 7 Intel 4-Core machines with i7-2600 CPU at 3.40GHz, 64 bits, 8GB RAM and 120 GB HDD Western Digital storage was used. Each traffic type is generated by using the models provided by LTE-Sim simulator adapted to generate UHD VR-based video large data packets.

The wireless channel is simulated by using the Jakes fast fading model, that is considered deterministic, similar to Rayleigh fading as it makes use of sinusoidal summing [31]. Jakes fading considers the central frequency of 2GHz, the system bandwidth in order to determine the periods of sinusoids, and the user speed to determine the pulsation and the number of paths for the initial phase calculation. In our case, the user speed is 3kmph with random direction in both learning and exploitation stages. Then, a number of 6 to 12 paths are randomly generated at each TTI as implemented in [36]. The channel propagation considers the loss given by: path, shadowing and penetration. We consider the urban microcell model for the path loss calculation, the shadowing loss is modelled as a log-normal distribution ($\mu = 0, \sigma = 8$ dB) in the range of [0, 20] dB, and the penetration loss is fixed to 10dB as it considers only the wall attenuation.

At each TTI, the user CQI is reported by following five steps. In the first step, the reference signal is broadcasted at each TTI by the base station over the entire system bandwidth. In the second step, each user calculates the power of the received reference signal that is attenuated by fading and propagation loss models. In the third step, each user measures the channel gain or the Signal-to-Interference/Noise Ratio (SINR) for each RB based on the received power and interference values. In our model, the intra-cell interference is negligible while the inter-cell interference considers a cluster of 7 cells for each component carrier. The ML-based solution and other schedulers run only on the central cell of each cluster, while other cells provide the inter-cell interference levels. In the fourth step, the CQI value for each RB is determined based on mapping curves between SINR and BLock Error Rate (BLER), where the target BLER is 10% [31]. Finally, the fifth step involves the transmission of each user CQI to the base station via a separate uplink channel which is errorless in our case.

We consider downlink transmission with carrier aggregation with a bandwidth of 100 MHz ($B = 500$), a micro cell radius of 200m and the FDD transmission mode. The CQI reporting scheme is full-band and periodically sent at each TTI to each user. The packet scheduler works on the carrier component basis and makes use of separate entities for RLC functionalities, retransmission schemes and modulation/coding assignments. Each RLC entity works in acknowledged mode and considers a maximum number of 5 retransmissions for each data packet. Packets failing to get successfully transmitted within this period are declared lost. The user PLRs and rates are summed per each carrier component at each TTI.

Four traffic classes with different QoS profiles are considered for scheduling, such as: 20% UHD VR-based live video streaming ($p = 1$), 60% live conventional video ($p = 2$), 15% voice ($p = 3$) and 5% file transfer ($p = 4$) [1]. UHD VR-based video traffic is generated with a rate higher than 20Mbps, where the packet delay requirement is 10ms and the packet loss rate less than 10^{-3}. The conversational video traffic has a variable data rate with a mean of 1Mbps and more relaxed QoS profile. In the frequency domain, a mixture of scheduling rules is considered, such as PF, BF ($w_1 = 1.25, w_2 = 1.31 \cdot 10^{-5}$), EXP ($w_3 = 6$) and OPLF ($w_4 = 10$) functions as detailed in Section IV-C.

A. Learning Stage

In the learning stage, the number of users for each traffic class is randomly chosen in the given ratio at predefined time slots in order to increase the possibility of the actor-critic neural networks to experience as many as possible variants of instantaneous states from different space regions. Under these circumstances, the optimal configuration of both actor and critic NNs must be found in terms of the number of hidden layers L_H and hidden nodes N_l, $l = \{2, \ldots, L\}$. With a lower number of hidden layers and nodes, the actor NN may underfit the input data in the sense that some regions of the state space are not very well represented by the learnt non-linear function. On the other hand, a higher number of hidden layers and nodes may determine the neural networks to overfit the training data, in the sense that, the framework will also learn the noisy data. In both cases, the critic error starts to increase at a certain moment of time in the learning stage. In order to find the best options for the number of hidden layers and nodes, we simulated the learning stage in parallel for about 10^7 TTIs (with the same networking conditions) for each of the following group of configurations: ($N_l = 150; L_H = \{1, 3, 5\}$), ($N_l = 200; L_H = \{1, 3, 5\}$), ($N_l = 250; L_H = \{1, 3, 5\}$ and ($N_l = 300; L_H = \{1, 3, 5\}$). Table II presents the numerical results of these configurations in terms of the critic error and system complexity.

By monitoring the minimum error of a neural network over the learning stage, the over-fitting can be detected when increasing the number of hidden layers and nodes. For example, if the error decreases as the NN topology increases, then the system can learn better with the higher configuration. On the other side, if the minimum error increases as the NN topology size increases, then the over-fitting can appear and the system can learn better with the lower configuration. As seen in Table II for $N_l = 150$ hidden nodes, the minimum critic error gets lower as the critic NN configuration increases from $L_H = 1$ to $L_H = 3$ and gets higher when increasing the number of layers from $L_H = 3$ to $L_H = 5$. For the first set of results ($N_l = 150; L_H = \{1, 3, 5\}$) obtained with the same networking conditions, it can be concluded that above 450 hidden nodes ($\{L_H = 3; N_l = 150\}$), the risk of over-fitting becomes higher. For other three sets of results ($N_l = \{200, 250, 300\}$), it can be observed that the critic error increases as the number of hidden layers increases from $L_H = 1$ to $L_H = 5$. Although these four sets of simulations are not obtained with the same networking conditions, it can be concluded that the critic NN configurations with ($L_H = 1, N_l = \{150, 200, 250, 300\}$) and ($L_H = 3, N_l = 150$) can be used for the proposed ML-based scheduling solution. The same observations are respected for the actor NN, with the amendment that the over-fitting appears much later since the weights are not updated at each TTI due to the critic decision. For a higher topology, the over-fitting can cause poor QoS provisioning for UHD VR users as well as over-provisioning of other traffic classes.

TABLE II
LEARNING PERFORMANCE OF DIFFERENT CONFIGURATIONS OF NEURAL NETWORKS

No. Hidden Nodes (N_l)	No. Hidden Layers (L_H)	Minimum Critic Error (E_C)	Normalized Complexity Forward Prop.	Normalized Complexity Backward Prop.
150	1	0.0116691	0.06	0.64
	3	0.0114227	0.21	0.88
	5	0.0120037	0.39	1.2
200	1	0.0119183	0.07	0.65
	3	0.0122024	0.35	1.11
	5	0.0121528	0.67	1.67
250	1	0.0121407	0.08	0.68
	3	0.0125644	0.53	1.45
	5	0.0122383	0.98	2.31
300	1	0.00969642	0.09	0.69
	3	0.0106559	0.73	1.8
	5	0.0107797	1.37	3.06

Alongside the performance of the critic error, Table II presents the complexity analysis for the forward and backward propagation of both actor and critic NNs. The backward propagation includes here the error propagation from output to the input layers and the refinement of NN weights. We measure the normalized complexity as a ratio between the sum of additional time (in seconds) needed to back-propagate the errors through critic and actor NNs at each TTI averaged over the total learning time (in seconds). Note that the backward propagation complexity of actor NN is measured only when the critic error is $E_c \geq 0$. The normalized complexity for the forward propagation procedure of both actor and critic NNs is determined in a similar way by averaging over the learning stage the accumulated time needed to forward the states from input to the output layers at each TTI. As seen in Table II, the normalized complexity of both monitored processes increases as the NN topology includes higher number of hidden layers and nodes. When considering the complexity analysis for the most indicated NN configurations from the perspective of over-fitting, we observe that a topology of ($L_H = 3, N_l = 150$) requires 3.5 times more computational time to forward propagate the states through the actor and critic NNs when compared

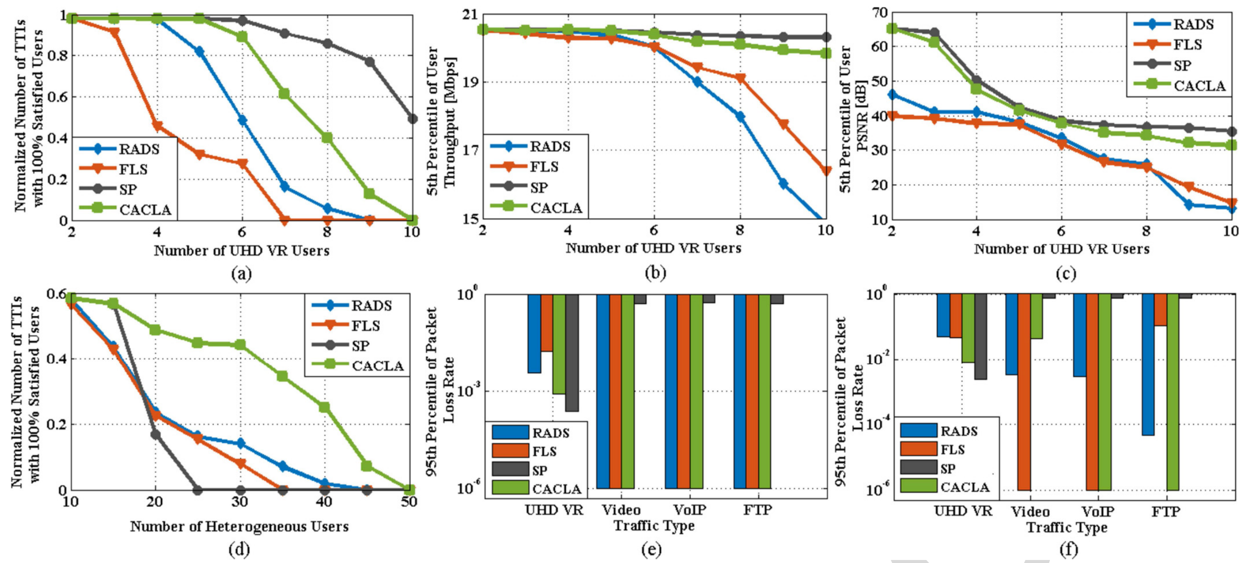


Fig. 4. (a) QoS provisioning (GBR, delay and PLR) for UHD VR-based live video streaming; (b) 5^{th} Percentile throughput performance for UHD VR-based live video streaming; (c) 5^{th} Percentile PSNR performance for UHD VR-based live video streaming; (d) Heterogeneous QoS provisioning (GBR, delay and PLR) for all traffic classes; (e) 95^{th} Percentile PLR performance per traffic type when the range of heterogeneous users is [10, 30]; (f) 95^{th} Percentile PLR performance per traffic type when the range of heterogeneous users is [31, 50].

to the case of ($L_H = 1, N_l = 150$). For the backward propagation, the normalized complexity ($L_H = 3, N_l = 150$) is only 1.5 times greater than that of ($L_H = 1, N_l = 150$) since the actor NN is not updated at each TTI. However, we are interested in exploiting the performance of the configuration that provides the lowest complexity ($L_H = 1, N_l = 150$). The additional execution overhead required by this configuration in the scheduling process is about 70% in the learning stage (6% for the forward propagation and 64% for the backward propagation) for both actor and critic neural networks. In the exploitation stage, the additional complexity is 3% since only the actor NN is used.

B. Exploitation Stage

In the exploitation stage, the performance of the proposed ML-based scheduling solution is analyzed when using the configuration of $L_H = 1$ and $N_l = 150$. The proposed CACLA framework is compared with FLS [20], RADS [21] and SP schemes. Among other scheduling approaches, RADS and FLS schedulers are time efficient and target a multitude of QoS objectives divided between time and frequency scheduling domains. The TP stage for FLS estimates the amount of real-time data to be transmitted in the next frame based on discrete-time linear control theory arguments. Then, the real-time flows are prioritized based on the approximated quota of data necessary to meet the delay constraints. The configuration details on this controlling loop can be found in [20]. The TP stage of RADS scheme is conducted based on a function that considers the fairness, delay and user rates in order to create an inter-class user prioritization at each TTI. The number of users to be passed to the FP scheduler at each TTI must be a priori configured. For our simulations, a maximum number of $U/2$ users show the best performance when measuring the average scheduling time when the heterogeneous QoS requirements are respected. For SP scheme, TP domain considers a static prioritization between different classes at each TTI as presented in Section IV-A. In the frequency domain, FLS employs the PF scheduler to improve the fairness between users preselected in the TP stage, whereas RADS and SP make use of the OPLF scheduler to enhance the PLR performance.

In order to measure the performance of the proposed solution in real time scheduling, three types of evaluations are considered: intra-class, aggregate and inter-class. For the intra-class evaluation (Figures 4.a, 4.b, 4.c), the aim is to measure the performance when scheduling the UHD VR-based live video traffic only. In this case, we evaluate the intra-class QoS provisioning, throughput and PSNR depending on U_1 number of UHD VR connections, where U_1 represents a ratio of 20% from the total number of heterogeneous users ($U_1 = 1/5 \cdot U$). The aggregate evaluation (Fig. 4.d) aims to measure the overall scheduling performance in terms of heterogeneous QoS provisioning as a function of the total number of active users U. The intra-class evaluation (Fig. 4.e and Fig. 4.f) presents the over-provisioning effect by considering the PLR performance of each scheduler per different traffic class. Finally, in Fig. 5 we analyze the execution overhead required by each scheduler while varying the number of heterogeneous users.

Figure 4.a presents the normalized scheduling duration when all QoS objectives (in terms of GBR, delay and PLR) are respected for the UHD VR-based live streaming traffic only. As expected, the SP scheme provides the highest possible performance as it gives the highest priority to the UHD VR-based live streaming traffic at all times. For the entire user range, CACLA performs much better than FLS and RADS by obtaining gains in excess of 100% when serving more than six UHD VR-based live video connections.

The Cumulative Distribution Function (CDF) of user throughput is determined at the end of the exploitation stage (for each configuration in terms of the number of users) based on the throughput values collected from each user at each TTI. Looking at the 5^{th} percentile of user throughput from the CDF curve (worst user throughput) for the UHD VR-based live streaming traffic (Fig. 4.b), smooth degradation can be observed in the case of CACLA scheme compared to SP when the number of UHD VR-based live streaming users goes above seven. When scheduling more than five users from the first class, RADS and FLS aim to focus more on scheduling lower priority users by degrading the user throughput of the first prioritized traffic class. As seen in Fig. 4.b, when scheduling eight UHD VR users, CACLA outperforms FLS and RADS by more than 1Mbps and 2Mbps, respectively. For ten users, the gain gets much higher at about 3Mbps and 5Mbps, respectively. This is because when the number of heterogeneous users gets very high, CACLA aims at working similarly to the SP scheme by providing a much higher prioritization to the UHD VR connections.

Figure 4.c presents the performance of the 5^{th} percentile PSNR in order to highlight the worst user PSNR performance when experiencing UHD VR content. This choice is motivated by the fact that PSNR is considered as one of the most popular objective QoE indicators used to evaluate the user perceived quality for video services [15]. Based on the evaluation provided in [37], an excellent Mean Opinion Score (MOS) can be obtained when $PSNR_{dB} \geq 36$ while an acceptable MOS is considered when $29 \leq PSNR_{dB} < 36$. Thus, a very good MOS performance for CACLA is obtained when scheduling less than eight users while an acceptable level can be attained for more than eight UHD VR users. When employing RADS and FLS schedulers, the best MOS performance is obtained for $U_1 \in [2, 5]$, an acceptable MOS value when $U_1 = 6$ and poor and even bad MOS levels are obtained when $U_1 > 6$. When $U_1 > 9$, CACLA obtains gains higher than 50% when compared to FLS and RADS in terms of the worst user PSNR.

When all the traffic classes are considered, we present in Fig. 4.d the performance when provisioning heterogeneous QoS. We monitor the number of TTIs when all users meet their QoS requirements by using the priority policies given by SP, RADS, FLS and CACLA. It can be noticed that SP is not able to provide an acceptable QoS level when scheduling more than 20 heterogeneous users. In this case, CACLA can achieve up to 50% more time when the heterogeneous QoS objectives are achieved. When reporting to RADS and FLS, CACLA can obtain gains higher than 100% for a range of scheduled users of $U \in [20, 40]$. When the number of users start to increase ($U > 45$), the achievement of QoS objectives gets close to the saturation. Consequently, CACLA aims to prioritize more the UHD VR traffic class as showing in Figures 4.b and 4.c.

For each traffic class, we monitor PLR values of each user at each TTI. At the end of each exploitation simulation, we compute the CDF curves for each of these classes in order to get the worst user percentiles of PLR. When compared to user throughput and PSNR, the worst PLR percentiles are found at the upper limit of the CDF curve. Figure 4.e analyses the inter-class performance when averaging the 95^{th} PLR percentiles for each traffic class over the range of $U \in [10, 30]$. When employing CACLA-based scheduling solution, up to 30 UHD VR connections can be supported (the PLR requirements are met) in the network while providing the requested PLR levels of other services. For this range, SP is over-provisioning the UHD VR traffic class being unable to assure the requested PLR for other traffic classes. RADS and FLS are unable to respect the PLR requirement of UHD VR traffic class (10^{-3}) when the worst user PLR is monitored.

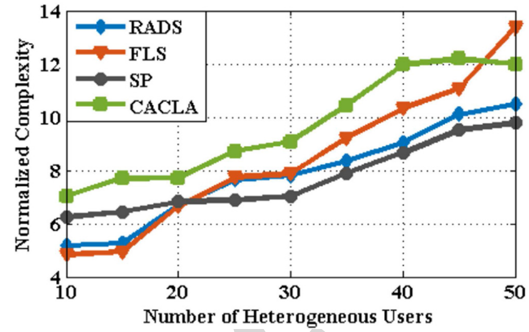


Fig. 5. System complexity of involved schedulers.

As stated previously, the RADS and FLS prioritization schemes are unable to react to the changeable networking conditions in terms of the number of active users U, variable arrival bit rates when generating the traffic, and wireless channel conditions. Thus, some traffic classes are over-provisioned while others may have degraded QoS performance. Figure 4.f demonstrates the aforementioned statement. The inter-class performance when averaging the 95^{th} PLR percentile for each traffic class over the range of $U \in [31, 50]$ is analyzed. This is achieved in order to monitor the behavior of each scheme when the heterogeneous QoS provisioning is getting closer to the saturation level due to the increase in number of users. As seen from this figure, FLS is over-provisioning the video and VoIP classes while degrading the QoS performance of the UHD VR-based live streaming traffic. As expected, the SP scheme prioritizes UHD VR users while drastically penalizing the rest of the traffic classes. CACLA prioritizes more the UHD VR-based live streaming class when the number of users is increasing, while it aims to give enhanced inter-class fairness when the number of users is lower and the QoS provisioning can be attained for each class as shown in Fig. 4.e. This is possible due to the adaptation capability of this policy when the number of users increases/decreases. The impact of the scheduling rule adaptability based on channel conditions and application characteristics is highlighted in Fig. 4.e, where CACLA is able to obtain better PLR performance than FLS and RADS while the PLR requirements for other classes are respected by all these candidates. The RADS scheme shows a notable limitation in Fig. 4.f due to the prioritization scheme used in time domain. A certain level of inter-class fairness can be observed but at lower PLR levels when compared to CACLA, even if the PLR minimization is considered in the frequency domain since the OPLF scheduler is employed.

Figure 5 represents the complexity analysis of the previously analyzed scheduling schemes. The complexity analysis measures the number of clock ticks elapsed for the TP and FP stages divided to the total number of clocks within one second and averaged over the exploitation stage duration (in seconds).

Below twenty aggregate users, FLS and RADS are less time consuming since the frequency domain scheduling is performed for a less number of users than that of SP and CACLA schemes. Since the networking conditions permit, CACLA and SP perform the FP stage for all four traffic classes. However, a slight complexity increase is required by the traffic class selection procedure when performing CACLA scheduling. Above this level of 20 aggregate users, SP solution gets the lowest complexity since only the first prioritized class (live UHD VR video users) is sent to the FP domain (see correlation with Fig. 4.a and Fig. 4.d.). Starting from the level of 30 heterogeneous users, RADS becomes a better option than FLS since the TP stage pre-selects a lower number of users to be sent in the frequency domain. At this point, RADS and FLS provide a complexity gain of 11.1% when compared to CACLA. As seen from Fig. 4.d, in the range of [30, 40] users, CACLA obtains gains in excess of 100% in terms of heterogeneous QoS provisioning when compared to FLS and RADS. However, this performance comes at the expense of the complexity increase as depicted in Fig. 5. Since the FP stage is performed for all traffic classes at almost each TTI, CACLA needs additional time resources in proportion of 20% to complete its tasks when compared to FLS, while the extra complexity requirement exceeds 30% when compared to RADS. Above this level, the complexity required by CACLA starts to stabilize or even to decrease since it behaves more like a SP scheme, while the FLS complexity becomes higher.

C. Practical Implications

According to our findings, some aspects must be considered when employing a RL-based scheduling solution for traffic prioritization, user scheduling and resource allocation in practice, such as: the training data set, the state space pre-processing, the controller configuration and termination condition for the learning stage. In order to get a generalised training data set, the training samples must consider variable number of users and changed at certain time intervals for each traffic class. Moreover, different speed levels and direction models should be considered for mobile users in order to explore a high variety of channel conditions. Under its original form, the training data-set is multidimensional and variable, depending on the number of active users that may change over time. Therefore, some pre-processing methods are necessary to compress the dimension of input state to some constant representations. Statistical methods can be used to get the mean and standard deviation values for the QoS indicators (i.e., packet loss, delay, throughput, etc.) for each traffic class [18]. Also, supervised learning can be used to classify the CQI reports in given patterns for users of each traffic class [31]. The optimal configuration of RL controller depends on the number of traffic classes and scheduling rules. When the number of traffic classes increases, higher number of hidden layers and nodes can be required with respect to some complexity constraints. Additionally, the output layer for the actor neural network must be properly managed and decoded in traffic class and scheduling rule selection as the size of the action space increases. During learning, both critic and actor errors must be monitored. In case of over-fitting (error increases above given threshold), the weights should be saved and learning process stopped. Otherwise, learning can continue for a number of iterations (TTIs) a priori established.

VI. Conclusion

This paper proposes an intelligent Machine Learning-based scheduling solution which makes use of Reinforcement Learning by employing CACLA, to react to the changeable networking conditions and take the best decisions in order to improve the fraction of time (in TTIs) when the QoS requirements are met for diverse services. Thus, the algorithm decides at each TTI the traffic class prioritization and the type of scheduling rule to be employed. Different traffic classes are dynamically prioritized such that the over-provisioning effect for some applications is avoided, whereas radio resources are intelligently managed by choosing the best scheduling rule for user scheduling and resource allocation. The proposed solution is deployed in a very challenging dynamic environment in which UAV performs UHD VR-based live video streaming to ground users. The proposed solution was evaluated through simulations and compared against other three state-of-the-art scheduling algorithms, such as: SP, RADS and FLS. The simulation results indicate that the proposed CACLA-based RL scheduling solution outperforms the other schemes involved while considering four perspectives: a) CACLA outperforms RADS and FLS in terms of packet loss, delay, throughput and PSNR when considering UHD VR-based users only; b) when considering a mixture of users requesting heterogeneous services, CACLA shows gains in excess of 100% by measuring the fraction of TTIs when the heterogeneous QoS requirements are respected; c) by measuring the inter-class packet loss, CACLA can accommodate a higher number of UHD VR users in the network, while SP and FLS prioritization schemes are over-provisioning some traffic classes; d) CACLA provides the best performance vs. complexity tradeoff.

Acknowledgment

G.-M. Muntean would like to acknowledge the Science Foundation Ireland grant 13/RC/2094 to Lero—the Irish Software Research Centre (http://www.lero.ie).

References

[1] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*, Cisco, San Jose, CA, USA, Feb. 2017. Accessed: Dec. 7, 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 133–144.

[3] L. Fay, L. Michael, D. Gómez-Barquero, N. Ammar, and M. W. Caldwell, "An overview of the ATSC 3.0 physical layer specification," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 159–171, Mar. 2016.

[4] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: Overview and Rel-11 enhancements," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 68–74, Nov. 2012.

[5] J. J. Gimenez *et al.*, "5G new radio for terrestrial broadcast: A forward-looking approach for NR-MBMS," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 356–368, Jun. 2019.

[6] X. Wang *et al.*, "Millimeter wave communication: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1616–1653, 3rd Quart., 2018.

[7] C. Ge *et al.*, "QoE-assured live streaming via satellite backhaul in 5G networks," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 381–391, Jun. 2019.

[8] A. Doumanoglou *et al.*, "A system architecture for live immersive 3D-media transcoding over 5G networks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 11–15.

[9] N. Jawad *et al.*, "Smart television services using NFV/SDN network management," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 404–413, Jun. 2019.

[10] R. Viola, A. Martin, M. Zorrilla, and J. Montalbán, "MEC proxy for efficient cache and reliable multi-CDN video distribution," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 1–7.

[11] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[12] A. Martin *et al.*, "Network resource allocation system for QoE-aware delivery of media services in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 561–574, Jun. 2018.

[13] E. Mangina, E. O'Keeffe, J. Eyerman, and L. Goodman, "Drones for live streaming of visuals for people with limited mobility," in *Proc. 22nd Int. Conf. Virtual Syst. Multimedia (VSMM)*, Oct. 2016, pp. 1–6.

[14] A. Doumanoglou *et al.*, "Quality of experience for 3-D immersive media streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 379–391, Jun. 2018.

[15] R. Trestian, I.-S. Comşa, and M. F. Tuysuz, "Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?" *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 945–977, 2nd Quart., 2018.

[16] N. D. Tripathi and J. H. Reed. *5G Evolution: On the Path to 6G Expanding the Frontiers of Wireless Communications White Paper, Rohde & Schwarz Technical Document 2019*, Accessed: Nov. 6, 2019. [Online]. Available: https://www.mobilewirelesstesting.com/beyond-5g-intro-6g/

[17] M. Hosseini, Y. Jiang, R. R. Berlin, L. Sha, and H. Song, "Toward physiology-aware DASH: Bandwidth-compliant prioritized clinical multimedia communication in ambulances," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2307–2321, Oct. 2017.

[18] I.-S. Comşa, A. De-Domenico, and D. Ktenas, "QoS-driven scheduling in 5G radio access networks—A reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. GLOBECOM*, Dec. 2017, pp. 1–7.

[19] S. Abedi, "Efficient radio resource management for wireless multimedia communications: A multidimensional QoS-based packet scheduler," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2811–2822, Nov. 2005.

[20] G. Piro, L. A. Grieco, G. Boggia, R. Fortuna, and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052–1065, Oct. 2011.

[21] G. Monghal, D. Laselva, P. Michaelsen, and J. Wigard, "Dynamic packet scheduling for traffic mixes of best effort and VoIP users in E-UTRAN downlink," in *Proc. IEEE 71st Veh. Technol. Conf.*, May 2010, pp. 1–5.

[22] X. Xie, X. Zhang, S. Kumar, and L. E. Li, "piStream: Physical layer informed adaptive video streaming over LTE," *GetMobile Mobile Comput. Commun.*, vol. 20, no. 2, pp. 31–34, Oct. 2016.

[23] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 308–318.

[24] I.-S. Comşa *et al.*, "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1661–1675, Aug. 2018.

[25] I.-S. Comşa, S. Zhang, M. Aydin, P. Kuonen, R. Trestian, and G. Ghinea, "A comparison of reinforcement learning algorithms in fairness-oriented OFDMA schedulers," *Information*, vol. 10, no. 10, p. 315, Oct. 2019.

[26] M. Elsayed and M. Erol-Kantarci, "AI-enabled radio resource allocation in 5G for URLLC and eMBB users," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Nov. 2019, pp. 590–595.

[27] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 94–101, Feb. 2018.

[28] *Technical Specification Group Services and System Aspects; Policy and Charging Control Architecture Release 12, v.12.2.0*, 3GPP, Sophia Antipolis, France, 2013.

[29] G. Song and Y. Li, "Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 127–134, Dec. 2005.

[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, U.K.: IMT Press, 2017.

[31] I.-S. Comşa, *Sustainable Scheduling Policies for Radio Access Networks Based on LTE Technology*, Ph.D. dissertation, Univ. Bedfordshire, Luton, U.K., 2014. AQ3

[32] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, R. Trestian, and G. Ghinea, "Guaranteeing user rates with reinforcement learning in 5G radio access networks," in *Next-Generation Wireless Networks Meet Advanced Machine Learning Applications*, vol. 8. Hershey, PA, USA: IGI Global, 2019, pp. 163–198.

[33] B. Sadiq, R. Madan, and A. Sampath, "Downlink scheduling for multiclass traffic in LTE," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, no. 14, pp. 1–18, 2009.

[34] N. Khan, M. Martini, Z. Bharucha, and G. Auer, "Opportunistic packet loss fair scheduling for delay-sensitive applications over LTE systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, vol. 1, Apr. 2012, pp. 1456–1461.

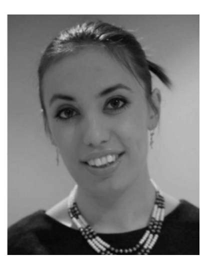
[35] H. Van Hasselt and M. Wiering, "Using continuous action spaces to solve discrete problems," in *Proc. Int. Joint Conf. Neural Netw.*, Apr. 2009, pp. 1149–1156.

[36] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498–513, Feb. 2011.

[37] A. Moldovan, I. Ghergulescu, and C. H. Muntean, "VQAMap: A novel mechanism for mapping objective video quality metrics to subjective MOS scale," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 610–627, Sep. 2016.

Ioan-Sorin Comşa received the B.Sc. and M.Sc. degrees in telecommunications from the Technical University of Cluj-Napoca, Romania, in 2008 and 2010, respectively, and the Ph.D. degree from the Institute for Research in Applicable Computing, University of Bedfordshire, U.K., in June 2015. He is a Research Scientist in 5G radio resource scheduling with Brunel University, London, U.K. He was a Ph.D. Researcher with the Institute of Complex Systems, University of Applied Sciences of Western Switzerland, Switzerland. Since 2015, he has been a Research Engineer with CEA-LETI, Grenoble, France. His research interests include intelligent radio resource and QoS management, reinforcement learning, data mining, distributed and parallel computing, and adaptive multimedia/mulsemedia delivery.

Gabriel-Miro Muntean (Senior Member, IEEE) is an Associate Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of the DCU Performance Engineering Laboratory. He has published over 350 papers in top-level international journals and conferences, authored four books and 18 book chapters, and edited seven additional books. His research interests include quality, performance, and energy issues related to rich media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and a reviewer for important international journals, conferences, and funding agencies. He is the Coordinator of the EU-funded project NEWTON http://www.newtonproject.eu.



Ramona Trestian (Member, IEEE) received the Ph.D. degree from Dublin City University, Ireland, in 2012. She is a Senior Lecturer with the Design Engineering and Mathematics Department, Middlesex University, London, U.K. She was with Dublin City University as an IBM/IRCSET Exascale Postdoctoral Researcher. She published in prestigious international conferences and journals and has five edited books. Her research interests include mobile and wireless communications, user perceived quality of experience, multimedia streaming, handover and network selection strategies, and digital twin modeling.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

PLEASE NOTE: We cannot accept new source files as corrections for your paper. If possible, please annotate the PDF proof we have sent you with your corrections and upload it via the Author Gateway. Alternatively, you may send us your corrections in list format. You may also upload revised graphics via the Author Gateway.

Carefully check the page proofs (and coordinate with all authors); additional changes or updates WILL NOT be accepted after the article is published online/print in its final form. Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections. Your article has been peer reviewed, accepted as final, and sent in to IEEE. No text changes have been made to the main part of the article as dictated by the editorial level of service for your publication.

AQ1: Please confirm or add details for any funding or financial support for the research of this article.
AQ2: Please provide the postal code for Dublin City University, Dublin 9, Ireland.
AQ3: Please provide the department name for Reference [31].

An Innovative Machine-Learning-Based Scheduling Solution for Improving Live UHD Video Streaming Quality in Highly Dynamic Network Environments

Ioan-Sorin Comşa, Gabriel-Miro Muntean, *Senior Member, IEEE*, and Ramona Trestian, *Member, IEEE*

***Abstract*—The latest advances in terms of network technologies open up new opportunities for high-end applications, including using the next generation video streaming technologies. As mobile devices become more affordable and powerful, an increasing range of rich media applications could offer a highly realistic and immersive experience to mobile users. However, this comes at the cost of very stringent Quality of Service (QoS) requirements, putting significant pressure on the underlying networks. In order to accommodate these new rich media applications and overcome their associated challenges, this paper proposes an innovative Machine Learning-based scheduling solution which supports increased quality for live omnidirectional (360°) video streaming. The proposed solution is deployed in a highly dynamic Unmanned Aerial Vehicle (UAV)-based environment to support immersive live omnidirectional video streaming to mobile users. The effectiveness of the proposed method is demonstrated through simulations and compared against three state-of-the-art scheduling solutions, such as: static Prioritization (SP), Required Activity Detection Scheduler (RADS) and Frame Level Scheduler (FLS). The results show that the proposed solution outperforms the other schemes involved in terms of PSNR, throughput and packet loss rate.**

***Index Terms*—Omnidirectional video, live streaming, QoS, machine learning, radio resource management, UAV.**

I. Introduction

GLOBAL mobile video traffic continues to grow exponentially, especially with the introduction of Ultra-High-Definition (UHD) or so called 4K video streaming applications. This new application category puts tremendous pressure on the current underlying networks as the average bit rate for 4K video is around 15 to 18Mbps, which is more than double the High Definition (HD) video bit rate and nine times more than the Standard Definition (SD) video bit rate [1].

Additionally, the increasing adoption of new Virtual Reality (VR) and Augmented Reality (AR) enabled high-end mobile devices together with the increasing amount of content ready to be consumed pushes the current 4G networks closer to their saturation. It is expected that the VR/AR generated traffic to continue to follow a high growth trajectory especially with the potential adoption of virtual reality streaming [1] that opens up a new era of 5G-based media services. Moreover, Cisco [1] also predicts that live Internet video will account for 17% of the Internet video traffic by 2022 with IP video traffic reaching 82% of all IP traffic globally.

Manuscript received December 6, 2019; revised March 9, 2020; accepted
AQ1 March 12, 2020. *(Corresponding author: Gabriel-Miro Muntean.)*
Ioan-Sorin Comşa is with the Department of Computer Science, Brunel University London, Uxbridge UB8 3PH, U.K. (e-mail: ioan-sorin.comsa@brunel.ac.uk).
AQ2 Gabriel-Miro Muntean is with the School of Electronic Engineering, Dublin City University, Dublin 9, Ireland (e-mail: gabriel.muntean@dcu.ie).
Ramona Trestian is with the Department of Design Engineering and Mathematics, Middlesex University, London NW4 4BT, U.K. (e-mail: r.trestian@mdx.ac.uk).
Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/TBC.2020.2983298

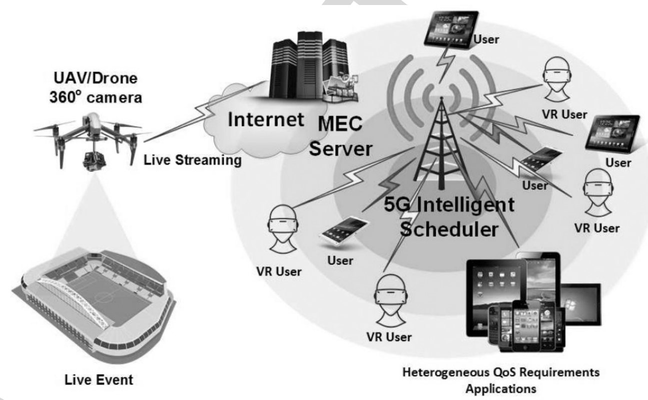


Fig. 1. Highly dynamic immersive live UHD streaming example scenario.

Consequently, in order to keep up with the current and predicted traffic demands, the network operators have already started an accelerated roll-out of 5G communications. As the new 5G technology targets high data rate and very low latency, it opens up a new range of applications starting from immersive augmented reality to driverless cars or even robot-enabled remote surgery. According to Cisco, by 2022, 5G devices and connections will represent more than 3% of global mobile devices and connections, with 12% of the global mobile traffic being generated over the 5G cellular network [1]. However, the network operators need to demonstrate that the tremendous potential of the 5G deployment could meet the users' expectations. The challenge is magnified even further especially given the current wide and diverse range of applications with different Quality of Service (QoS) requirements which need to be supported on a heterogeneity of end-user hardware platforms. Applications such as live network streaming require low latency and jitter, whereas, reliability is needed for applications such as file transfer which cannot tolerate packet loss or high delay. As most applications require end-to-end network support, this hampers the potential development and advantages of new applications. Consequently, it becomes obvious that just increasing the system capacity is not enough to meet the heterogeneous QoS requirements for all mobile users at the same time. This is mainly due to the increasing popularity of bandwidth-hungry applications (e.g., multimedia-based applications), limited radio resources and changeable wireless network conditions. Thus, along with the next generation networks deployment, new emerging technologies and solutions are being explored to help network operators to cope with such high traffic demands, such as: integration of MPEG-DASH [2] as the de-facto video delivery mechanism, Advanced Television Systems Committee (ATSC) 3.0 standard [3], evolved Multimedia Broadcast/Multicast Service (eMBMS) [4], Further eMBMS (FeMBMS) and New Radio MBMS (NR-MBMS) [5], mmWave communications [6], satellite back-haul [7], Software Defined Networks (SDN) and Network Function Virtualisation (NFV) [8], [9], Mobile Edge Computing (MEC) [10], Unmanned Aerial Vehicle (UAV) or drones [11], machine learning [12], etc. As a potential use case of UAV, Mangina *et al.* [13] make use of drones for live streaming for people with limited mobility, so that they could enjoy the immersion as if they were present at the specific location. The aim of this framework is to use the technology to enable opportunities for communication and self expression of people of all levels of physical and cognitive ability.

This work focuses on a highly dynamic mobile scenario involving high bitrate live video streaming, as the one illustrated in Fig. 1. In this scenario, an UAV equipped with an omnidirectional (360°) camera is used to send 4K/8K video captured in real time from a live event taking place for instance in a stadium, to a MEC server attached to a 5G network. VR-enabled users get the live video stream served via the 5G network and expect to enjoy a high quality video experience, as if they were present at the venue. However, to be able to create a high quality immersive experience for the remote users, the network operators need to guarantee low latency and packet loss, and high throughput while also accommodating other traffic classes. Unfortunately, this is not possible to achieve with conventional resource management methods.

In this context, this paper proposes and describes an innovative Machine Learning (ML)-based scheduling solution for radio resource management to improve significantly QoS provisioning and increase users' Quality of Experience (QoE) levels in the presence of heterogeneous traffic. The proposed solution targets particularly highly challenging scenarios which involve live streaming of very high bitrate video in highly dynamic network environments.

The remainder of this article is organized as follows: Section II discusses important related works in this area and Section III presents an overview of the proposed solution. Section IV details the proposed innovative ML-based scheduling solution for increased quality of live high bitrate video streaming in highly dynamic network environments and presents the associated problem formulation. Evaluation results are discussed in Section V in comparison with those of alternative solutions and finally, conclusions are drawn in Section VI.

II. Related Works

A key challenge for network operators is to provide ubiquitous connectivity to different device types and applications with heterogeneous QoS requirements. This challenge is amplified by the increasing popularity of multimedia-based bandwidth-hungry applications with strict QoS requirements that stretch the current 4G networks closer to saturation. Consequently, to be able to accommodate all these new immersive live streaming applications, known for being bandwidth-hungry and having low-latency and packet loss requirements [14], advanced solutions must be adopted to maintain increased QoE for end-users, since QoE is expected to become the biggest differentiator between network operators [15].

An important component that is expected to be integrated within the 5G and beyond 5G networks is the use of UAV [16]. Apart from facilitating temporary radio access and Internet connectivity, UAVs could also be used to facilitate live video broadcasting and enable support for high data rate transmissions [11]. However, to accommodate a high number of users with enhanced QoE levels within the 5G radio access network, system bandwidth needs to be properly managed. According to [17], two adaptation methods classes can be considered to deal with the bandwidth efficiency in order to improve QoS and QoE, such as: passive and active. The active approaches aim to improve the bandwidth allocation by using scheduling algorithms, whereas passive ones refer more to bandwidth-compliant adaptation techniques that adapt the multimedia transmission to the available bandwidth.

As an active adaptation entity, the packet scheduler is responsible for dynamically sharing the system bandwidth between the end-users such that the QoS provisioning is maximized. Different scheduling strategies are proposed in the literature to deal with QoS targets [18]. A scheduler that encapsulates the features of different scheduling strategies is proposed in [19] for 3G downlink systems to assure the multidimensional QoS provisioning under varying traffic and radio channel conditions. However, most of the state-of-the-art schedulers targeting multidimensional QoS requirements aim to prioritize some traffic classes while ignoring others. For instance, Frame Level Scheduler (FLS) [20] prioritizes real-time traffic (e.g., video, voice, gaming) over the more elastic traffic classes (e.g., file transfer, HTTP). In contrast, Required Activity Detection (RADS) [21] prioritizes a group of users according to their packet delay and fairness criterion. However, most of the prioritization schemes are unable to react to the dynamics of the wireless environment, such as: increasing number of users, various traffic characteristics, and changeable network conditions. As a consequence, some traffic classes are over-provisioned while others may have a degraded QoS.

A passive method used for traffic prioritization and bandwidth adaptation is proposed in [17] to manage the transmission of massive clinical applications in high-speed ambulance scenario under variable and limited communication bandwidth.

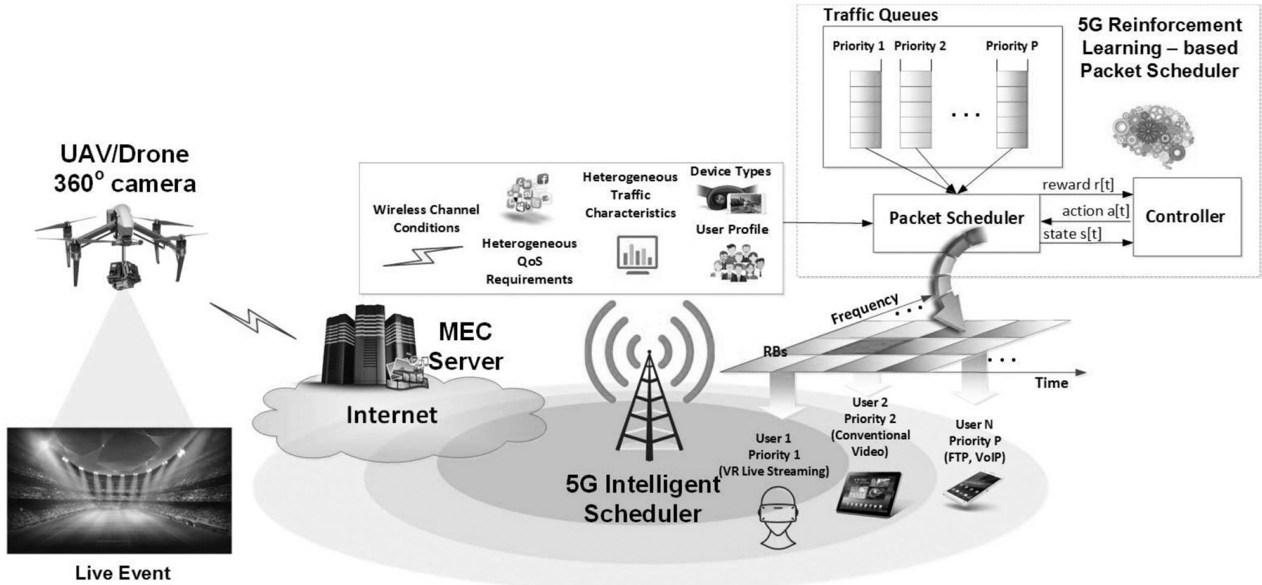


Fig. 2. Proposed 5G UAV-based live streaming framework.

The approach works in two stages: a) the clinical multimedia data is prioritized in four classes based on the disease model and the criticality of each model; b) according to the available bandwidth, different heuristic algorithms are proposed to reduce the clinical data rates according to their priority class. The evaluations show the effectiveness of this approach by transferring the most critical information within the limited bandwidth. By focusing only on QoE improvement, the system bandwidth can remain underutilized. In this sense, a passive adaptation scheme is proposed in [22] to facilitate the video rate adaptation by considering the physical layer information to enable accurate bandwidth estimation. The latest network advancements need to accommodate advanced applications and services with very high data rates and extremely low latency. Wang *et al.* [23] propose the use of fog networking to coordinate a network of drones equipped with cameras to broadcast live events. The objective of the proposed framework is to maximizing the coverage area as well as the available throughput for high-quality video streaming to video servers.

In terms of Radio Resource Management (RRM) and QoS provisioning, classical RRM functionalities would not be able to meet the stringent QoS requirements of all these immersive live streaming applications while also catering for the rest of application classes. In the context of 5G, ML is currently gaining considerable attention as it is seen as one of the key enablers for QoS provisioning [12], [18], [24]–[26] as well as for the development of intelligent services for smart cities [27]. An autonomous network resource management for QoS and QoE provisioning is proposed in [12] to predict the amount of network resources that needs to be allocated to cope with the traffic demands for live and on-demand dynamic adaptive streaming over HTTP. Machine learning is used to optimize the scheduling and resource allocation problems in 5G radio access networks focusing on different combinations of QoS objectives, such as: throughput, delay and packet loss in [18], packet loss and delay in [24], system throughput and user fairness in [25]. However, these ML-based scheduling solutions are designed for homogeneous traffic types only. The ML framework proposed in [26] aims to optimize the resource and power allocation problem for heterogeneous traffic with the scope of improving the delay of Ultra-Reliable and Low-Latency Communications (URLLC) users and throughput of enhanced Mobile Broadband (eMBB) users. Compared to previous works, this paper proposes a ML-based scheduling and resource allocation solution to enable high level of QoS provisioning for mobile users experiencing UAV VR-based live video content while maintaining an acceptable service quality of other traffic types with diverse QoS requirements.

To this extent, the contributions of this paper are two fold:

- an innovative ML-based scheduling solution to enable QoS provisioning for Ultra High Definition video streaming in highly dynamic network environments;
- a QoS-oriented UAV-based integrated system for enabling high quality levels for immersive live video streaming.

The benefits of the proposed ML-based solution compared to other state-of-the-art schedulers are summarized as follows:

- enhanced QoS provisioning (in terms of delay, throughput and packet loss requirements), higher throughput and Peak Signal-to-Noise Ratio (PSNR) for users requesting UHD VR-based live video;
- gains in excess of 100% when monitoring the time fraction when the heterogeneous QoS requirements are met in a mixture of services with various QoS requirements;
- improved inter-class fairness by respecting over time the standard prioritization order; it can accommodate a higher number of UHD VR video connections and avoids the over/under-provisioning of other traffic classes.

III. Proposed Framework for UAV-Based 4K Streaming

The main components of the proposed quality and performance-oriented system for high quality live video streaming are illustrated in Fig. 2. The figure presents a very challenging deployment involving a UAV with a 360° camera, a MEC server, a 5G intelligent packet scheduler and VR users. The UAV has a 360° spherical camera that records a live event (e.g., football games, concerts, festivals, etc.). The UAV communicates via the 5G network on the ground to send 4K/8K UHD video to the MEC server. For simplicity, it is assumed that there is no loss on the communication link between the UAV and the MEC server. The MEC server will then stream live the UHD video content to the users. However, in order to accommodate a heterogeneous traffic mix with different QoS requirements, an intelligent ML-based packet scheduler is proposed to enable high QoS provisioning for different traffic classes, including for live high bitrate video streaming. The mix of traffic can consider the 5G services and use cases such as eMBB, URLLC and massive Machine Type Communications (mMTC) as well as other types of 4G related services with more relaxed QoS requirements.

The role of the packet scheduler is to allocate the available frequency resources to active users within a given cell to improve as much as possible the fraction of scheduling time when the QoS requirements are met for each traffic type. The scheduling process is conducted at each Transmission Time Interval (TTI) and usually works in two steps: a) Time-based Prioritization (TP) where a group of users with more stringent QoS requirements is prioritized among other users with more relaxed QoS constraints and b) Frequency-based Prioritization (FP) that aims to allocate the radio resources in order to increase the QoS provisioning in terms of delay, packet loss and rate requirements for the pre-selected group of users. While time prioritization is seen as an outer QoS provisioning scheme for all traffic classes based on a given priority order, frequency prioritization acts as an inner QoS provisioning scheme for the pre-selected users. Consequently, the scheduler will prioritize data packets in both time and frequency domains based on current networking conditions that may change at each TTI, including: number of users for each traffic class, QoS profiles, heterogeneous QoS parameters, VR live streaming characteristics, channel conditions, etc. However, many existing scheduling schemes are not able to adapt to the dynamic and unpredictable networking conditions [18]. For instance, some time-based prioritization schemes aim to over-provision some traffic classes while degrading the performance of others [20], [21], whereas the frequency-based prioritization techniques will address only particular QoS requirements at any time [18]. In order to avoid these drawbacks, the proposed scheduling solution is flexible, being able to adapt according to the current network conditions in order to enhance the fraction of time when the heterogeneous QoS requirements are respected.

Since live UHD VR-based video streaming has strict QoS requirements with data rates at least twenty times greater than other conventional applications [1], the best practice would be to decide at each TTI the most suitable traffic class to be prioritized in order to: a) meet the very stringent QoS requirements of live UHD VR-based traffic and b) avoid the starvation effect for other types of applications. In the frequency domain, the most suitable scheduling rule is selected to improve the QoS provisioning for each selected traffic class. Therefore, an intelligent ML-based solution is introduced to learn over time and propose the most suitable prioritization decisions based on current scheduler states. Therefore, this paper proposes an innovative ML-based scheduler for heterogeneous traffic in Orthogonal Frequency Division Multiple Access (OFDMA) downlink systems. The proposed ML-based scheduling solution is able to take each time two scheduling decisions in order to increase the amount of time when all QoS requirements are met. This two-dimensional decision prioritizes a certain traffic class at each TTI and decides the scheduling rule that allocates the available bandwidth to users of the pre-selected class in the frequency domain.

IV. Intelligent ML-Based Scheduling Solution

As previously stated, the proposed ML-based scheduler (see Fig. 2) is able to select at each TTI the most suitable traffic class to be prioritized in time domain and the best scheduling rule for the user prioritization in frequency domain in order to improve the QoS provisioning. These decisions could be taken based on various parameters, such as: wireless channel conditions, application requirements, traffic characteristics, users' profile, device types, etc. The details of the ML-based scheduler are presented next in this section.

A. Prioritization-Based Scheduling

In frequency domain, it is considered that the available bandwidth is divided in equal Resource Blocks (RBs), the smallest radio resource that can be allocated by the Base Station (BS) to the user (see Fig. 2). We define by $\mathcal{B} = \{1, 2, \ldots, B\}$ the set of available RBs in a given bandwidth. To get the necessary bandwidth needed to accommodate a high number of UHD VR-enabled live video streaming connections, we aggregate multiple radio bandwidths. Each User Equipment (UE) is characterized by a single traffic class, with a given priority and a QoS profile in terms of delay, packet loss and throughput requirements. Multiple UEs may request different services with heterogeneous QoS requirements. A successful scheduler should be able to accommodate UHD VR-based live services as well as other conventional traffic types (e.g., video, voice, file transfer, etc) without penalizing one over the other. The list of symbols used in this paper is presented in Table I.

Let us consider P the number of traffic classes with different QoS profiles. We define by $\mathcal{P} = \{1, 2, \ldots, P\}$ the priority set such that traffic class 1 has the highest priority (i.e., UHD VR-based live streaming traffic) while traffic class P has the lowest priority. The *Static prioritization (SP)* is defined according to the 3GPP guidelines [28] as follows: regardless of the network conditions, the scheduling process respects the priority set $\mathcal{P} = \{1, 2, \ldots, P\}$ for the entire downlink transmission session. Let us define the set of active users for all classes as $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_P\}$, where $\mathcal{U}_p$ is the subset of users corresponding to traffic class $p \in \mathcal{P}$. We denote by U_p the number of users belonging to class $p \in \mathcal{P}$, while by U, the total number of active users from all classes. Moreover, the set of heterogeneous QoS objectives in terms of their requirements' accomplishment is defined as $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_P\}$,

TABLE I
LIST OF NOTATIONS

Parameter	Description
$\mathcal{A}$	Discrete and two-dimensional controller action space
$\mathbf{a}[t]$	Current action $\mathbf{a} \in \mathcal{A}$ decided at TTI t
$\mathcal{B}$	Set of resource blocks from different carriers
b	Random resource block $b \in \mathcal{B}$
B	Max. no. of resource blocks
E_c	Error of critic neural network
E_a	Error of actor neural network
L_H	Number of hidden layers
$m_{b,p,u}$	Metric of user $u \in \mathcal{U}_p$ on RB $b \in \mathcal{B}$
N_l	Number of nodes corresponding to layer l
$\mathcal{O}$	Set of heterogeneous objectives
$\mathcal{O}_p$	Set of objectives corresponding to class p
o	Objective index belonging to a given set $\mathcal{O}_p$
O_p	Number of QoS objectives for the traffic class $p \in \mathcal{P}$
$\mathcal{P}$	Set of traffic classes in the priority order given by [28]
p	Random traffic class $p \in \mathcal{P}$
P	Max. no. of traffic classes
$\mathcal{R}$	Set of scheduling rules
r	Random scheduling rule $r \in \mathcal{R}$
R	Max. no. of scheduling rules from $\mathcal{R}$
$\mathcal{S}$	Continuous and multi-dimensional scheduler state space
$\mathbf{s}[t]$	Current scheduler state $\mathbf{s} \in \mathcal{S}$ at TTI t
$\mathcal{U}$	Set of heterogeneous users
$\mathcal{U}_p$	Set of users corresponding to class p
u	User index belonging to a given class $\mathcal{U}_p$
U_p	Number of active users from $\mathcal{U}_p$
U	Total number of heterogeneous users
$x_{o,p,u}$	QoS indicator of $o \in \mathcal{O}$ and user $u \in \mathcal{U}_p$
$\bar{x}_{o,p,u}$	QoS requirement of $o \in \mathcal{O}$ and user $u \in \mathcal{U}_p$
$\Gamma_{d,u}$	Utility function of rule $r \in \mathcal{R}$ and user $u \in \mathcal{U}_p$
$\rho[t+1]$	System reward value received at TTI t+1

where $\mathcal{O}_p$ is the set of objectives for class $p \in \mathcal{P}$. It is said that set $\mathcal{O}_p$ is met if the delay, packet loss and throughput requirements are respected by all active users belonging to traffic class $p \in \mathcal{P}$.

In frequency domain, the process of user scheduling and resource allocation is conducted according to a given scheduling rule that is oriented on a particular QoS objective or on a group of QoS objectives. We define the set of scheduling rules as $\mathcal{R} = \{1, 2, \ldots, R\}$, where R represents the maximum number of rules. Assuming that a SP scheme is employed at this stage at each TTI, the set of active users $\mathcal{U}_1$ is passed in the frequency domain for scheduling. Here, a given scheduling rule $r \in \mathcal{R}$ contributes to the metric computation for each user $u \in \mathcal{U}_1$ on each RB $b \in \mathcal{B}$. Each metric shows how necessary is for each user $u \in \mathcal{U}_1$ to get each resource $b \in \mathcal{B}$ from the perspective of the addressed objective $o \in \mathcal{O}_1$ targeted by the scheduling rule $r \in \mathcal{R}$. In the initial phase of scheduling, a number of U_1 metrics is computed for each RB $b \in \mathcal{B}$ by summing a total number of $U_1 \cdot B$ metrics. In the second phase, the scheduler allocates each RB $b \in \mathcal{B}$ to the user with the highest metric and the process is repeated RB-by-RB until the entire set $\mathcal{B}$ is allocated. However, some metrics can be zero since the QoS objectives are met or there are not enough packets in the queue for some users. If all metrics are equal, then the RB $b \in \mathcal{B}$ remains unoccupied. Finally, the third phase of the scheduling process aims at calculating the size of the transport block for each user scheduled on different RBs and determines the modulation and coding scheme necessary to decode the data at the reception. The scheduling process can be repeated for the next prioritized class (i.e., $p = 2$) if some RBs are unoccupied once the users from $\mathcal{U}_1$ are scheduled.

By employing this SP scheme, the UHD VR-based live video streaming traffic is always allocated the best resources while adversely affecting QoS provisioning for other traffic classes. To avoid this fundamental drawback, other traffic classes must be prioritized when network conditions are favorable. Consequently, in this work, the proposed approach aims to select at each TTI the traffic class $p \in \mathcal{P}$ in such a way that the satisfaction of heterogeneous QoS requirements has the highest possible outcome under the current networking conditions. In this way, we decide at each TTI the prioritization set $\mathcal{P}[t] = \{p, 1, \ldots, p-1, p+1, \ldots, P\}$, where class $p \in \mathcal{P}$ gets as many resources as needed up to the maximum number of RBs, whereas other classes receive the remaining resources by following the priority order of $\{1, \ldots, p-1, p+1, \ldots, P\}$. Even so, if always applying the same scheduling rule for frequency prioritization, only one objective across all traffic classes would be addressed, while harming the performance of other QoS targets. Consequently, in the frequency domain, our aim is to apply at each TTI the most suitable scheduling rule in order to increase the fraction of time (in TTIs) when the heterogeneous QoS requirements are met.

B. Multi-Class and Multi-Objective Optimization Problem

Let us define by $x_{p,u,o}$ the Key Performance Indicator (KPI) of user $u \in \mathcal{U}_p$ and objective $o \in \mathcal{O}_p$ and by $\bar{x}_{p,u,o}$ its associated requirement. It is said that user $u \in \mathcal{U}_p$ meets objective $o \in \mathcal{O}_p$ if and only if $x_{p,u,o}$ respects $\bar{x}_{p,u,o}$. Furthermore, let us define the current KPI vector $\mathbf{x}_{p,u}[t] = [x_{p,u,o_1}, x_{p,u,o_2}, \ldots, x_{p,u,O_p}]$ and its associated requirement vector $\bar{\mathbf{x}}_{p,u} = [\bar{x}_{p,u,o_1}, \bar{x}_{p,u,o_2}, \ldots, \bar{x}_{p,u,O_p}]$. User $u \in \mathcal{U}_p$ meets all QoS objectives if and only if $\mathbf{x}_{p,u}$ respects the requirement vector $\bar{\mathbf{x}}_{p,u}$. By extending this reasoning, the entire set of objectives is met for each traffic class $p \in \mathcal{P}$, if vector $\mathbf{x}_p[t] = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \ldots, \mathbf{x}_{p,U_p}]$ respects its requirements $\bar{\mathbf{x}}_p = [\bar{\mathbf{x}}_{p,1}, \bar{\mathbf{x}}_{p,2}, \ldots, \bar{\mathbf{x}}_{p,U_p}]$. The proposed framework aims to increase the number of TTIs when the KPI vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_P]$ respects the QoS requirement vector $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_P]$. We formulate in (1) the multi-class and multi-objective optimization problem that aims to determine at each TTI the most convenient traffic class to be prioritized and scheduling rule to be applied in the frequency domain such that vector of QoS indicators $\mathbf{x}$ reaches the highest possible outcome when reporting to the vector of QoS requirements $\bar{\mathbf{x}}$.

$$\max_{i,j,k} \sum_{r\in\mathcal{R}} \sum_{p\in\mathcal{P}} \sum_{u\in\mathcal{U}_p} \sum_{b\in\mathcal{B}} i_{r,p}[t] \cdot j_{p,u}[t] \cdot k_{u,b}[t] \cdot \Gamma_{r,p}(\mathbf{x}_{p,u}[t]) \times \gamma_{u,b}[t], \tag{1}$$

$$s.t. \quad \sum_{u} k_{u,b}[t] \leq 1, \quad b = 1, \ldots, B, \tag{1a}$$

$$\sum_{p} j_{p,u}[t] \leq 1, \quad u = u_1, \ldots, u_{U_p}, p = 1, \ldots, P, \tag{1b}$$

$$\sum_{u} j_{p^*,u}[t] = U_{p^*}, \quad p^* \in \mathcal{P}, \tag{1c}$$

$$\sum_{u} j_{p^{\otimes},u}[t] = 0, \quad \forall p^{\otimes} \in \mathcal{P} \backslash \{p^*\}, \tag{1d}$$

$$\sum_{r} i_{r,p}[t] = 1, \quad p = 1, 2, \ldots, P, \tag{1e}$$

$$\sum_{p} i_{r^*,p}[t] = P, \quad r^* \in \mathcal{R}, \tag{1f}$$

$$\sum_{p} i_{r^{\otimes},p}[t] = 0, \quad \forall r^{\otimes} \in \mathcal{R}\backslash\{r^*\}, \tag{1g}$$

$$i_{r,p}[t] \in \{0, 1\}, \quad \forall r \in \mathcal{R}, \forall p \in \mathcal{P}, \tag{1h}$$

$$j_{p,u}[t] \in \{0, 1\}, \quad \forall p \in \mathcal{P}, \forall u \in \mathcal{U}_p, \tag{1i}$$

$$k_{u,b}[t] \in \{0, 1\}, \quad \forall u \in \mathcal{U}_p, \forall b \in \mathcal{B}. \tag{1j}$$

In (1) $\gamma_{u,b}[t]$ is the achievable user rate that quantifies the number of bits transmitted if the RB $b \in \mathcal{B}$ would be allocated to user $u \in \mathcal{U}_p$. Basically, $\gamma_{u,b}[t]$ is determined based on the Channel Quality Indicator (CQI), a bandwidth dependent vector reported by each user $u \in \mathcal{U}_p$ to the base station. For each scheduling rule $r \in \mathcal{R}$, a unique utility function $\Gamma_{r,p}(\mathbf{x}_{p,u})$ is associated in order to attenuate the channel variations given by $\gamma_{u,b}[t]$ and to provide to the user the priority to be scheduled in the frequency domain. Any utility function $\Gamma_{r,p}(\mathbf{x}_{p,u}) : \mathbf{R} \rightarrow \mathbf{R}$ must be monotone and concave [29]. The utility functions can be designed in many ways by considering different KPIs as arguments with certain impact when meeting the heterogeneous and multidimensional QoS requirements. More examples of utility functions are presented in the next section. When setting the same utility function $\Gamma_{r,p}(\mathbf{x}_{p,u})$ for all traffic classes, no matter what the prioritization set $\mathcal{P}_p[t]$ is, the KPI vector $\mathbf{x}$ respects the requirement vector $\bar{\mathbf{x}}$ in a certain measure. The idea is to select at each TTI the prioritization set $\mathcal{P}_p[t]$ and the most suitable utility such that the QoS provisioning would be maximized.

The traffic class, scheduling rule and radio resources are assigned based on the decision variables. In (1), $k_{u,b}[t]$ is the resource allocation variable: $k_{u,b}[t] = 1$ when RB $b \in \mathcal{B}$ is allocated to UE $u \in \mathcal{U}_p$ and $k_{u,b}[t] = 0$, otherwise. Constraints in (1a) aim to allocate at most one user to each RB. Variable $j_{p,u}[t]$ assigns each user to a specific traffic class. Constraints (1b) indicate that each user belongs to at most one traffic class. Constraints (1c) and (1d) show that only users from the selected traffic class $p^* \in \mathcal{P}$ are passed in the frequency domain. Variable $i_{r,p}[t]$ determines the type of utility to be selected at each TTI. Constraints (1e) indicate that one type of utility function per traffic class is selected at each TTI, whereas constraints (1f) and (1g) show that the same scheduling rule is selected for all traffic classes, where variable $r^* \in \mathcal{R}$ is the selected scheduling rule at TTI t and $r^{\otimes} \in \mathcal{R}$ are the other scheduling rules remained un-selected at TTI t. Constraints (1h), (1i) and (1j) make the entire problem combinatorial.

Due to very high complexity, solving the optimization problem from (1) at each TTI is difficult to achieve. Thus, we propose a sub-optimal solution aiming to split this problem in two sub-problems: in the first sub-problem, the prioritization set $\mathcal{P}_p[t]$ is decided and the most appropriated scheduling rule $r \in \mathcal{R}$ is assigned; in the second sub-problem, the resource allocation is performed based on the prioritized traffic class and selected scheduling rule. For the first sub-problem, we propose a ML-based approach [30] to decide at each TTI the class $p^* \in \mathcal{P}$ to be prioritized at first and the best fitting scheduling rule $r^* \in \mathcal{R}$ for the resource allocation. The second sub-problem aims to solve the user scheduling from $\mathcal{U}_{p^*}$ and the resource allocation based on the selected scheduling rule $r^* \in \mathcal{R}$ as described in Section IV-A. As a first step of the scheduling process, we determine the metric $m_{b,p^*,u}$ for each user $u \in \mathcal{U}_{p^*}$ and RB $b \in \mathcal{B}$ at each TTI as follows:

$$m_{b,p^*,u}[t] = \Gamma_{r^*,p^*}\big(\mathbf{x}_{p^*,u}\big) \cdot \gamma_{u,b}[t]. \tag{2}$$

As a result, the matrix of metrics $\mathbf{m} = [m_{b,p^*,u}] \in \mathbb{R}^{U_{p^*} \times B}$ is computed, where $b = \{1, 2, \ldots, B\}$ and $u = \{u_1, u_2, \ldots, u_{U_{p^*}}\}$. For each RB $b \in \mathcal{B}$, a vector of metrics is considered, such as: $\mathbf{m}_b = [m_{b,p^*,u_1}, m_{b,p^*,u_2}, \ldots, m_{b,p^*,u_{U_{p^*}}}]$. Resource $b \in \mathcal{B}$ is allocated to that user that has the maximum metric value from the vector $\mathbf{m}_b$, written in the following manner:

$$b \mapsto u, \quad if \quad u = argmax_{u'}\big(m_{b,p^*,u'}[t]\big), \tag{3}$$

where expression $b \mapsto u$ allocates RB b to user u and $k_{u,b} = 1$. It is important to mention that the allocation is performed RB-by-RB until the entire set of RBs $\mathcal{B}$ gets allocated. However, if for example $\mathbf{m}_{b'} = [0, 0, \ldots, 0]$, then RB $b' \in \mathcal{B}$ remains unoccupied. This resource can be allocated when the scheduling process is repeated for the next prioritized traffic class from the remained set of $\mathcal{P}[t] \setminus \{p^*\}$. By following this model, under certain network conditions it might happen that not all the users could get enough resources to meet their QoS objectives. The aim of the proposed scheduler is to increase as much as possible the QoS provisioning for UHD VR video users with insignificant QoS degradation of other services by properly selecting each time the traffic class to be prioritized and the scheduling rule to be performed in the frequency domain.

C. Types of Scheduling Rules

A scheduling rule $r \in \mathcal{R}$ provides a unique utility function $\Gamma_{r,p}(\mathbf{x}_{p,u})$ focused on a particular or a group of QoS objectives. User fairness is one of the most popular objectives which can be addressed when employing the following function [31]:

$$\Gamma_{1,p}\big(\bar{T}_{p,u}\big) = 1/\bar{T}_{p,u} \tag{4}$$

where $\bar{T}_{p,u}$ is the average throughput of user $u \in \mathcal{U}_p$ calculated based on the exponential moving filter and the scheduling rule $r = 1$ is Proportional Fair (PF). According to (2), (3) and (4), user $u \in \mathcal{U}_p$ with the highest ratio between achievable rate and average throughput on RB $b \in \mathcal{B}$ is selected, while keeping a certain fairness with the previously served users.

Guaranteeing the Bit Rates (GBR) is another QoS objective that can be addressed when selecting the function [32]:

$$\Gamma_{2,p}\Big(\bar{\bar{T}}_{p,u}\Big) = [1 + w_1 \cdot e^{-w_2 \cdot \left(\bar{\bar{T}}_{p,u} - T^R_{p,u}\right)}] \cdot \Gamma_{1,p}\big(\bar{T}_{p,u}\big). \tag{5}$$

where $\bar{\bar{T}}_{p,u}$ is the average user throughput calculated with the median moving filter and $r = 2$ is the Barrier Function (BF) scheduling rule. Users with lower average rates than that of the corresponding requirements $T^R_{p,u}$ are preferred to be scheduled on each RB.

Delay objective aims at respecting the Head-of-Line (HoL) packet delay of each user at each TTI. One possible solution to achieve this target is to employ the following function [33]:

$$\Gamma_{3,p}\big(D_{p,u}\big) = e^{w_3 \cdot D_{p,u}/D^R_{p,u}} \cdot \Gamma_{1,p}\big(\bar{T}_{p,u}\big), \tag{6}$$

where $D_{p,u}$ is the HOL delay of user $u \in \mathcal{U}_p$ at TTI t, $D^R_{p,u}$ is the corresponding requirement and $r = 3$ is entitled the EXPonential (EXP) rule. Users with packets approaching to their deadline receive a much higher priority to be scheduled given the exponential function.

The Packet Loss Rate (PLR) of each user can be improved when the scheduler employs the following utility function [34]:

$$\Gamma_{4,p}\big(L_{p,u}\big) = w_4 \cdot L_{p,u}/L^R_{p,u} \cdot \Gamma_{1,p}\big(\bar{T}_{p,u}\big), \tag{7}$$

where $L_{p,u}$ is the PLR value at TTI t of user $u \in \mathcal{U}_p$, $L^R_{p,u}$ is the corresponding PLR requirement and $r = 4$ is the Opportunistic Packet Loss Fair (OPLF) scheduling rule. When the throughput, delay and PLR requirements are met by all users, BF, EXP and OPLF, respectively act similar to the PF scheduling rule.

D. Controller and Packet Scheduler Interaction

In order to increase the fraction of scheduling time when the heterogeneous QoS requirements are respected, we propose the use of Reinforcement Learning (RL) [30] to learn the most suitable traffic prioritization and scheduling rule that can be applied in real time scheduling. RL makes use of an agent (e.g., intelligent controller) that in time will learn to take actions which will generate the maximum reward by interacting with the environment (e.g., packet scheduler). As seen from Fig. 2, at TTI t, the controller observes a state $\mathbf{s}[t] \in \mathcal{S}$, representing the current network conditions, and takes an action $\mathbf{a}[t] = [p, r] \in \mathcal{A}$ that prioritizes traffic class $p \in \mathcal{P}$ in time domain and selects the scheduling rule $r \in \mathcal{R}$ to be applied in the frequency domain. The scheduling procedure is conducted based on the selected action and the system evolves to the next state $\mathbf{s}[t+1] = \mathbf{s}' \in \mathcal{S}$ at TTI $t+1$. As illustrated in Fig. 2, the reward value received from the scheduling environment evaluates the performance of the applied action in the previous state. This function is calculated based on the set of KPIs $\mathbf{x}[t+1] = \mathbf{x}'$ received at TTI $t+1$. If we define the reward function as $\rho : \mathcal{X} \to [-1, 1]$, where $\mathcal{X} \subset \mathcal{S}$ is the state space of KPI vectors, then the proposed function takes the following form:

$$\rho\big(\mathbf{s}'\big) = \sum_p \sum_o w_p \cdot \rho_{p,o}\Big(\mathbf{x}'_p\Big), \tag{8}$$

where $\rho_{p,o}$ is the reward value of traffic class $p \in \mathcal{P}$ and objective $o \in \mathcal{O}_p$, respectively. In (8), $\mathbf{x}'_p$ is the KPI vector of class $p \in \mathcal{P}$ at TTI t+1. This $\rho_{p,o}$ value denotes how far the online KPI parameters of traffic class $p \in \mathcal{P}$ are from their requirements in terms of objective $o \in \mathcal{O}_p$. The weight w_p sets the 3GPP priority for each class as denoted by the static prioritisation set $\mathcal{P}$. The controller must explore a high number of state-to-state transitions to optimize the prioritization decisions.

E. RL-Based Scheduling Framework

Since the scheduler state space is multi-dimensional and continuous, the scheduling problems cannot be enumerated exhaustively. We can only approximate the best traffic class to be prioritized and the scheduling rule to be performed in the frequency domain, such that the QoS provisioning is much improved. To reduce the complexity for the learning framework, Neural Network (NN) is used to approximate the best prioritization decisions at each current state. During the learning stage, the NN weights are updated at each TTI based on the scheduler and controller interaction as shown in Fig. 2. In the exploitation stage, these weights are saved and the neural network is implemented as a non-linear function.

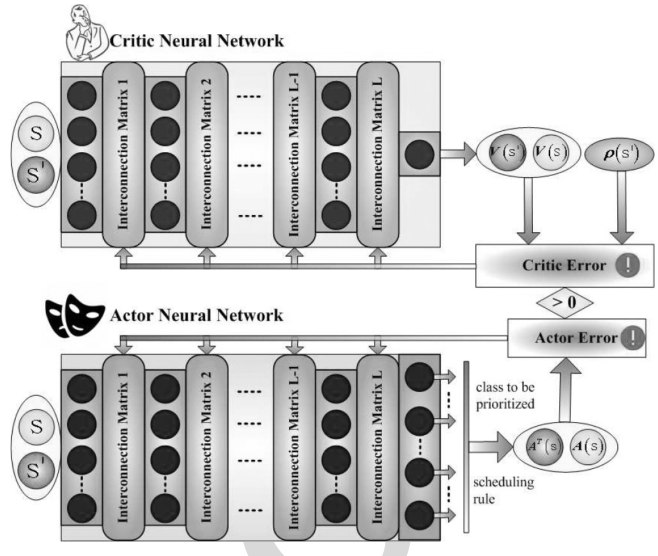


Fig. 3. CACLA-based RL controller architecture.

We propose the implementation of RL framework with a minimum complexity. In this sense, let M be the number of NN output pins in which, the first $M/2$ pins can be used to determine the index of the traffic class to be prioritized and the rest of output pins to decide the scheduling rule to be applied in the frequency domain. To train this non-linear function with multi-dimensional input and output variables, we use Continuous Actor-Critic Learning Automata (CACLA) algorithm [35]. As seen from Fig. 3, CACLA considers two neural networks: a) the critic neural network that approximates the state value function and criticizes the action taken on each state; b) actor neural network that approximates the best prioritization set $\mathcal{P}_p[t]$ and scheduling rule $r \in \mathcal{R}$ to be applied on each state. The role of the critic function is to examine the actor activity and improve its decisions over time.

As an internal structure, a neural network is composed by L number of layers, including here the hidden and output layers only. Therefore, we define the number of hidden layers as $L_H = L - 1$. Each layer $l \in \{1, 2, \ldots, L+1\}$ is composed by neurons or nodes and interconnection matrices that represent the weights connecting the nodes within two consecutive layers, for example l and $l+1$. If N_l and N_{l+1} are the number of nodes (not including the bias nodes) of layers l and $l+1$, respectively, then the total number of weights to be updated at each TTI is $\sum_{l=1}^{L} (N_l + 1) \cdot N_{l+1}$. As indicated in Fig. 3, when CACLA algorithm is employed, two sets of weights need to be updated since both actor and critic neural networks are involved during the learning stage.

The functional structure of critic NN is taking the form of the non-linear function defined as: $V : \mathcal{S} \rightarrow [-1, 1]$. The actor NN takes the same form with the amendment that the output value is multi-dimensional and the definition domain is $A : \mathcal{S} \rightarrow [-1, 1]^M$. In the learning stage, two steps are performed at each TTI: a) the updating step in which the weights of both neural networks V and A are updated according to CACLA algorithm and b), the action selection step, that determines the policy of how the controller action is selected at each TTI. In the exploitation stage, only the learnt actor function is used to provide the M dimensional decision under the form of the controller action $\mathbf{a}[t + 1] = [p, r]$ that can be decoded into traffic class prioritization and scheduling rule selection.

The updating process based on CACLA algorithm aims to refine the weights of both networks iteratively, on each state. For example, when the current state is $\mathbf{s}' \in \mathcal{S}$, the error between the impact of applied action $\mathbf{a}[t] \in \mathcal{A}$ in the previous state $\mathbf{s}[t] \in \mathcal{S}$ and its expectation must be reinforced through the neural networks. Since CACLA makes use of two neural networks, then two types of errors must be reinforced.

Critic Error: At the beginning of the learning stage, the weights of the critic NN are randomly chosen. Thus, these weights are gradually updated based on the quality of the applied actions in every state. As seen in Fig. 3, the adaptation of the critic NN weights comprises two steps: a) *forward propagation* responsible to get the consecutive critic values $\{V(\mathbf{s}), V(\mathbf{s}')\} \in [-1, 1]$ in order to quantify the impact of action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{s} \in \mathcal{S}$; and b) *back-propagation* step that calculates the critic error and propagates it through the critic NN based on the gradient descent principle [35]. Without going into details, the gradient descent calculates the error for each neuron of each layer $l \in \{2, \ldots, L + 1\}$ and updates the weights accordingly. The critic error function $E_c : \mathcal{S} \times \mathcal{S} \rightarrow [-1, 1]$ is defined (9), where $\{V^T(\mathbf{s}), V(\mathbf{s})\}$ are determined by propagating the states $(\mathbf{s}, \mathbf{s}')$ through the critic NN from input to the output layers:

$$E_c(\mathbf{s}', \mathbf{s}) = V^T(\mathbf{s}) - V(\mathbf{s}). \tag{9}$$

Here, the target value is determined as $V^T(\mathbf{s}) = \rho + \gamma \cdot V(\mathbf{s}')$, where $\gamma \in [0, 1]$ is a discount factor and ρ is the reward value calculated with (8).

Actor Error: If the critic error is positive $E_c(\mathbf{s}', \mathbf{s}) \geq 0$, then the previous action was a good choice and the actor NN can be updated as well. If $E_c(\mathbf{s}', \mathbf{s}) < 0$, then the previous action was an unfortunate choice and then, the actor NN must be discouraged in taking such decision in the future. Consequently, the actor NN is not updated. When $E_c(\mathbf{s}', \mathbf{s}) \geq 0$, the actor NN is updated by following the same forward and backward propagation principles. The multi-dimensional actor error is determined based on the function $E_a : \mathcal{S} \rightarrow [-1, 1]^M$:

$$E_a(\mathbf{s}) = A^T(\mathbf{s}) - A(\mathbf{s}), \tag{10}$$

where A^T is the target multi-dimensional action value determined based on some probability distributions. At the beginning of the learning stage, it is not recommended to exploit the actor NN decisions and then, a random multi-dimensional value of $A^T(\mathbf{s})$ different from $A(\mathbf{s})$ is preferred in order to enlarge the exploration of the scheduler state space. This is denoted as the *improvement* step. Once the learning process is approaching to its deadline, we aim to exploit more the actor decisions and then, the multi-dimensional target $A^T(\mathbf{s})$ is equal to $A(\mathbf{s})$. This is denoted as the *exploitation* step. For an optimal learning, it is preferred to mix improvement and exploitation steps with certain probabilities. Certainly, more improvements steps are preferred at the beginning of the learning stage, whereas the end of the learning stage is likely to use more exploitation steps. In this way, we monitor if the mean actor error can converge or not to certain error levels. Once the neural network(s) is(are) updated, the RL controller decides the new action $\mathbf{a}' \in \mathcal{A}$ to be applied in state $\mathbf{s}' \in \mathcal{S}$.

V. System Evaluation

The proposed adaptation framework was implemented in the RRM Scheduler Simulator [31], which is a C/C++ object oriented tool that inherits the LTE-Sim simulator [36]. For the performance evaluation, an infrastructure of 7 Intel 4-Core machines with i7-2600 CPU at 3.40GHz, 64 bits, 8GB RAM and 120 GB HDD Western Digital storage was used. Each traffic type is generated by using the models provided by LTE-Sim simulator adapted to generate UHD VR-based video large data packets.

The wireless channel is simulated by using the Jakes fast fading model, that is considered deterministic, similar to Rayleigh fading as it makes use of sinusoidal summing [31]. Jakes fading considers the central frequency of 2GHz, the system bandwidth in order to determine the periods of sinusoids, and the user speed to determine the pulsation and the number of paths for the initial phase calculation. In our case, the user speed is 3kmph with random direction in both learning and exploitation stages. Then, a number of 6 to 12 paths are randomly generated at each TTI as implemented in [36]. The channel propagation considers the loss given by: path, shadowing and penetration. We consider the urban microcell model for the path loss calculation, the shadowing loss is modelled as a log-normal distribution ($\mu = 0, \sigma = 8$ dB) in the range of [0, 20] dB, and the penetration loss is fixed to 10dB as it considers only the wall attenuation.

At each TTI, the user CQI is reported by following five steps. In the first step, the reference signal is broadcasted at each TTI by the base station over the entire system bandwidth. In the second step, each user calculates the power of the received reference signal that is attenuated by fading and propagation loss models. In the third step, each user measures the channel gain or the Signal-to-Interference/Noise Ratio (SINR) for each RB based on the received power and interference values. In our model, the intra-cell interference is negligible while the inter-cell interference considers a cluster of 7 cells for each component carrier. The ML-based solution and other schedulers run only on the central cell of each cluster, while other cells provide the inter-cell interference levels. In the fourth step, the CQI value for each RB is determined based on mapping curves between SINR and BLock Error Rate (BLER), where the target BLER is 10% [31]. Finally, the fifth step involves the transmission of each user CQI to the base station via a separate uplink channel which is errorless in our case.

We consider downlink transmission with carrier aggregation with a bandwidth of 100 MHz ($B = 500$), a micro cell radius of 200m and the FDD transmission mode. The CQI reporting scheme is full-band and periodically sent at each TTI to each user. The packet scheduler works on the carrier component basis and makes use of separate entities for RLC functionalities, retransmission schemes and modulation/coding assignments. Each RLC entity works in acknowledged mode and considers a maximum number of 5 retransmissions for each data packet. Packets failing to get successfully transmitted within this period are declared lost. The user PLRs and rates are summed per each carrier component at each TTI.

Four traffic classes with different QoS profiles are considered for scheduling, such as: 20% UHD VR-based live video streaming ($p = 1$), 60% live conventional video ($p = 2$), 15% voice ($p = 3$) and 5% file transfer ($p = 4$) [1]. UHD VR-based video traffic is generated with a rate higher than 20Mbps, where the packet delay requirement is 10ms and the packet loss rate less than 10^{-3}. The conversational video traffic has a variable data rate with a mean of 1Mbps and more relaxed QoS profile. In the frequency domain, a mixture of scheduling rules is considered, such as PF, BF ($w_1 = 1.25, w_2 = 1.31 \cdot 10^{-5}$), EXP ($w_3 = 6$) and OPLF ($w_4 = 10$) functions as detailed in Section IV-C.

A. *Learning Stage*

In the learning stage, the number of users for each traffic class is randomly chosen in the given ratio at predefined time slots in order to increase the possibility of the actor-critic neural networks to experience as many as possible variants of instantaneous states from different space regions. Under these circumstances, the optimal configuration of both actor and critic NNs must be found in terms of the number of hidden layers L_H and hidden nodes N_l, $l = \{2, \ldots, L\}$. With a lower number of hidden layers and nodes, the actor NN may underfit the input data in the sense that some regions of the state space are not very well represented by the learnt non-linear function. On the other hand, a higher number of hidden layers and nodes may determine the neural networks to overfit the training data, in the sense that, the framework will also learn the noisy data. In both cases, the critic error starts to increase at a certain moment of time in the learning stage. In order to find the best options for the number of hidden layers and nodes, we simulated the learning stage in parallel for about 10^7 TTIs (with the same networking conditions) for each of the following group of configurations: ($N_l = 150; L_H = \{1, 3, 5\}$), ($N_l = 200; L_H = \{1, 3, 5\}$), ($N_l = 250; L_H = \{1, 3, 5\}$ and ($N_l = 300; L_H = \{1, 3, 5\}$). Table II presents the numerical results of these configurations in terms of the critic error and system complexity.

By monitoring the minimum error of a neural network over the learning stage, the over-fitting can be detected when increasing the number of hidden layers and nodes. For example, if the error decreases as the NN topology increases, then the system can learn better with the higher configuration. On the other side, if the minimum error increases as the NN topology size increases, then the over-fitting can appear and the system can learn better with the lower configuration. As seen in Table II for $N_l = 150$ hidden nodes, the minimum critic error gets lower as the critic NN configuration increases from $L_H = 1$ to $L_H = 3$ and gets higher when increasing the number of layers from $L_H = 3$ to $L_H = 5$. For the first set of results ($N_l = 150; L_H = \{1, 3, 5\}$) obtained with the same networking conditions, it can be concluded that above 450 hidden nodes ($\{L_H = 3; N_l = 150\}$), the risk of over-fitting becomes higher. For other three sets of results ($N_l = \{200, 250, 300\}$), it can be observed that the critic error increases as the number of hidden layers increases from $L_H = 1$ to $L_H = 5$. Although these four sets of simulations are not obtained with the same networking conditions, it can be concluded that the critic NN configurations with ($L_H = 1, N_l = \{150, 200, 250, 300\}$) and ($L_H = 3, N_l = 150$) can be used for the proposed ML-based scheduling solution. The same observations are respected for the actor NN, with the amendment that the over-fitting appears much later since the weights are not updated at each TTI due to the critic decision. For a higher topology, the over-fitting can cause poor QoS provisioning for UHD VR users as well as over-provisioning of other traffic classes.

TABLE II
LEARNING PERFORMANCE OF DIFFERENT CONFIGURATIONS OF NEURAL NETWORKS

No. Hidden Nodes (N_l)	No. Hidden Layers (L_H)	Minimum Critic Error (E_C)	Normalized Complexity Forward Prop.	Normalized Complexity Backward Prop.
150	1	0.0116691	0.06	0.64
	3	0.0114227	0.21	0.88
	5	0.0120037	0.39	1.2
200	1	0.0119183	0.07	0.65
	3	0.0122024	0.35	1.11
	5	0.0121528	0.67	1.67
250	1	0.0121407	0.08	0.68
	3	0.0125644	0.53	1.45
	5	0.0122383	0.98	2.31
300	1	0.00969642	0.09	0.69
	3	0.0106559	0.73	1.8
	5	0.0107797	1.37	3.06

Alongside the performance of the critic error, Table II presents the complexity analysis for the forward and backward propagation of both actor and critic NNs. The backward propagation includes here the error propagation from output to the input layers and the refinement of NN weights. We measure the normalized complexity as a ratio between the sum of additional time (in seconds) needed to back-propagate the errors through critic and actor NNs at each TTI averaged over the total learning time (in seconds). Note that the backward propagation complexity of actor NN is measured only when the critic error is $E_c \geq 0$. The normalized complexity for the forward propagation procedure of both actor and critic NNs is determined in a similar way by averaging over the learning stage the accumulated time needed to forward the states from input to the output layers at each TTI. As seen in Table II, the normalized complexity of both monitored processes increases as the NN topology includes higher number of hidden layers and nodes. When considering the complexity analysis for the most indicated NN configurations from the perspective of over-fitting, we observe that a topology of ($L_H = 3, N_l = 150$) requires 3.5 times more computational time to forward propagate the states through the actor and critic NNs when compared

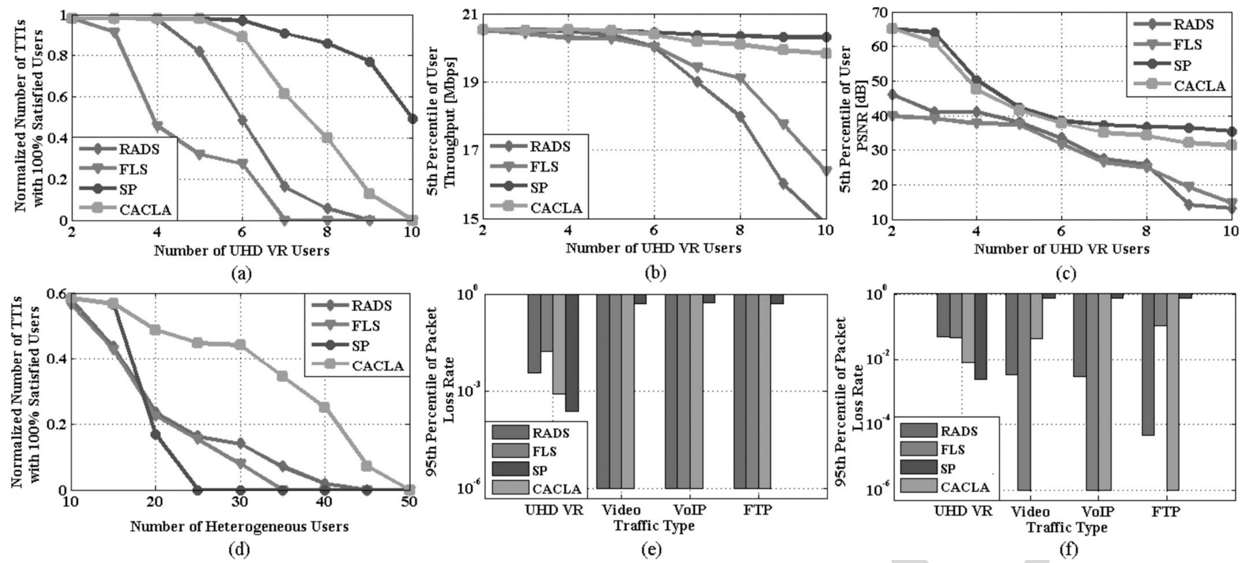


Fig. 4. (a) QoS provisioning (GBR, delay and PLR) for UHD VR-based live video streaming; (b) 5^{th} Percentile throughput performance for UHD VR-based live video streaming; (c) 5^{th} Percentile PSNR performance for UHD VR-based live video streaming; (d) Heterogeneous QoS provisioning (GBR, delay and PLR) for all traffic classes; (e) 95^{th} Percentile PLR performance per traffic type when the range of heterogeneous users is [10, 30]; (f) 95^{th} Percentile PLR performance per traffic type when the range of heterogeneous users is [31, 50].

to the case of ($L_H = 1, N_l = 150$). For the backward propagation, the normalized complexity ($L_H = 3, N_l = 150$) is only 1.5 times greater than that of ($L_H = 1, N_l = 150$) since the actor NN is not updated at each TTI. However, we are interested in exploiting the performance of the configuration that provides the lowest complexity ($L_H = 1, N_l = 150$). The additional execution overhead required by this configuration in the scheduling process is about 70% in the learning stage (6% for the forward propagation and 64% for the backward propagation) for both actor and critic neural networks. In the exploitation stage, the additional complexity is 3% since only the actor NN is used.

B. Exploitation Stage

In the exploitation stage, the performance of the proposed ML-based scheduling solution is analyzed when using the configuration of $L_H = 1$ and $N_l = 150$. The proposed CACLA framework is compared with FLS [20], RADS [21] and SP schemes. Among other scheduling approaches, RADS and FLS schedulers are time efficient and target a multitude of QoS objectives divided between time and frequency scheduling domains. The TP stage for FLS estimates the amount of real-time data to be transmitted in the next frame based on discrete-time linear control theory arguments. Then, the real-time flows are prioritized based on the approximated quota of data necessary to meet the delay constraints. The configuration details on this controlling loop can be found in [20]. The TP stage of RADS scheme is conducted based on a function that considers the fairness, delay and user rates in order to create an inter-class user prioritization at each TTI. The number of users to be passed to the FP scheduler at each TTI must be a priori configured. For our simulations, a maximum number of $U/2$ users show the best performance when measuring the average scheduling time when the heterogeneous QoS requirements are respected. For SP scheme, TP domain considers a static prioritization between different classes at each TTI as presented in Section IV-A. In the frequency domain, FLS employs the PF scheduler to improve the fairness between users preselected in the TP stage, whereas RADS and SP make use of the OPLF scheduler to enhance the PLR performance.

In order to measure the performance of the proposed solution in real time scheduling, three types of evaluations are considered: intra-class, aggregate and inter-class. For the intra-class evaluation (Figures 4.a, 4.b, 4.c), the aim is to measure the performance when scheduling the UHD VR-based live video traffic only. In this case, we evaluate the intra-class QoS provisioning, throughput and PSNR depending on U_1 number of UHD VR connections, where U_1 represents a ratio of 20% from the total number of heterogeneous users ($U_1 = 1/5 \cdot U$). The aggregate evaluation (Fig. 4.d) aims to measure the overall scheduling performance in terms of heterogeneous QoS provisioning as a function of the total number of active users U. The intra-class evaluation (Fig. 4.e and Fig. 4.f) presents the over-provisioning effect by considering the PLR performance of each scheduler per different traffic class. Finally, in Fig. 5 we analyze the execution overhead required by each scheduler while varying the number of heterogeneous users.

Figure 4.a presents the normalized scheduling duration when all QoS objectives (in terms of GBR, delay and PLR) are respected for the UHD VR-based live streaming traffic only. As expected, the SP scheme provides the highest possible performance as it gives the highest priority to the UHD VR-based live streaming traffic at all times. For the entire user range, CACLA performs much better than FLS and RADS by obtaining gains in excess of 100% when serving more than six UHD VR-based live video connections.

The Cumulative Distribution Function (CDF) of user throughput is determined at the end of the exploitation stage (for each configuration in terms of the number of users) based on the throughput values collected from each user at each TTI. Looking at the 5^{th} percentile of user throughput from the CDF curve (worst user throughput) for the UHD VR-based live streaming traffic (Fig. 4.b), smooth degradation can be observed in the case of CACLA scheme compared to SP when the number of UHD VR-based live streaming users goes above seven. When scheduling more than five users from the first class, RADS and FLS aim to focus more on scheduling lower priority users by degrading the user throughput of the first prioritized traffic class. As seen in Fig. 4.b, when scheduling eight UHD VR users, CACLA outperforms FLS and RADS by more than 1Mbps and 2Mbps, respectively. For ten users, the gain gets much higher at about 3Mbps and 5Mbps, respectively. This is because when the number of heterogeneous users gets very high, CACLA aims at working similarly to the SP scheme by providing a much higher prioritization to the UHD VR connections.

Figure 4.c presents the performance of the 5^{th} percentile PSNR in order to highlight the worst user PSNR performance when experiencing UHD VR content. This choice is motivated by the fact that PSNR is considered as one of the most popular objective QoE indicators used to evaluate the user perceived quality for video services [15]. Based on the evaluation provided in [37], an excellent Mean Opinion Score (MOS) can be obtained when $PSNR_{dB} \geq 36$ while an acceptable MOS is considered when $29 \leq PSNR_{dB} < 36$. Thus, a very good MOS performance for CACLA is obtained when scheduling less than eight users while an acceptable level can be attained for more than eight UHD VR users. When employing RADS and FLS schedulers, the best MOS performance is obtained for $U_1 \in [2, 5]$, an acceptable MOS value when $U_1 = 6$ and poor and even bad MOS levels are obtained when $U_1 > 6$. When $U_1 > 9$, CACLA obtains gains higher than 50% when compared to FLS and RADS in terms of the worst user PSNR.

When all the traffic classes are considered, we present in Fig. 4.d the performance when provisioning heterogeneous QoS. We monitor the number of TTIs when all users meet their QoS requirements by using the priority policies given by SP, RADS, FLS and CACLA. It can be noticed that SP is not able to provide an acceptable QoS level when scheduling more than 20 heterogeneous users. In this case, CACLA can achieve up to 50% more time when the heterogeneous QoS objectives are achieved. When reporting to RADS and FLS, CACLA can obtain gains higher than 100% for a range of scheduled users of $U \in [20, 40]$. When the number of users start to increase ($U > 45$), the achievement of QoS objectives gets close to the saturation. Consequently, CACLA aims to prioritize more the UHD VR traffic class as showing in Figures 4.b and 4.c.

For each traffic class, we monitor PLR values of each user at each TTI. At the end of each exploitation simulation, we compute the CDF curves for each of these classes in order to get the worst user percentiles of PLR. When compared to user throughput and PSNR, the worst PLR percentiles are found at the upper limit of the CDF curve. Figure 4.e analyses the inter-class performance when averaging the 95^{th} PLR percentiles for each traffic class over the range of $U \in [10, 30]$. When employing CACLA-based scheduling solution, up to 30 UHD VR connections can be supported (the PLR requirements are met) in the network while providing the requested PLR levels of other services. For this range, SP is over-provisioning the UHD VR traffic class being unable to assure the requested PLR for other traffic classes. RADS and FLS are unable to respect the PLR requirement of UHD VR traffic class (10^{-3}) when the worst user PLR is monitored.

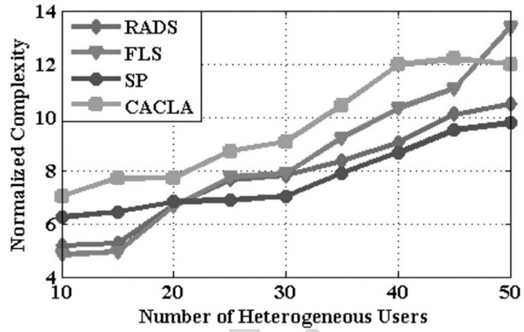


Fig. 5. System complexity of involved schedulers.

As stated previously, the RADS and FLS prioritization schemes are unable to react to the changeable networking conditions in terms of the number of active users U, variable arrival bit rates when generating the traffic, and wireless channel conditions. Thus, some traffic classes are over-provisioned while others may have degraded QoS performance. Figure 4.f demonstrates the aforementioned statement. The inter-class performance when averaging the 95^{th} PLR percentile for each traffic class over the range of $U \in [31, 50]$ is analyzed. This is achieved in order to monitor the behavior of each scheme when the heterogeneous QoS provisioning is getting closer to the saturation level due to the increase in number of users. As seen from this figure, FLS is over-provisioning the video and VoIP classes while degrading the QoS performance of the UHD VR-based live streaming traffic. As expected, the SP scheme prioritizes UHD VR users while drastically penalizing the rest of the traffic classes. CACLA prioritizes more the UHD VR-based live streaming class when the number of users is increasing, while it aims to give enhanced inter-class fairness when the number of users is lower and the QoS provisioning can be attained for each class as shown in Fig. 4.e. This is possible due to the adaptation capability of this policy when the number of users increases/decreases. The impact of the scheduling rule adaptability based on channel conditions and application characteristics is highlighted in Fig. 4.e, where CACLA is able to obtain better PLR performance than FLS and RADS while the PLR requirements for other classes are respected by all these candidates. The RADS scheme shows a notable limitation in Fig. 4.f due to the prioritization scheme used in time domain. A certain level of inter-class fairness can be observed but at lower PLR levels when compared to CACLA, even if the PLR minimization is considered in the frequency domain since the OPLF scheduler is employed.

Figure 5 represents the complexity analysis of the previously analyzed scheduling schemes. The complexity analysis measures the number of clock ticks elapsed for the TP and FP stages divided to the total number of clocks within one second and averaged over the exploitation stage duration (in seconds).

Below twenty aggregate users, FLS and RADS are less time consuming since the frequency domain scheduling is performed for a less number of users than that of SP and CACLA schemes. Since the networking conditions permit, CACLA and SP perform the FP stage for all four traffic classes. However, a slight complexity increase is required by the traffic class selection procedure when performing CACLA scheduling. Above this level of 20 aggregate users, SP solution gets the lowest complexity since only the first prioritized class (live UHD VR video users) is sent to the FP domain (see correlation with Fig. 4.a and Fig. 4.d.). Starting from the level of 30 heterogeneous users, RADS becomes a better option than FLS since the TP stage pre-selects a lower number of users to be sent in the frequency domain. At this point, RADS and FLS provide a complexity gain of 11.1% when compared to CACLA. As seen from Fig. 4.d, in the range of [30, 40] users, CACLA obtains gains in excess of 100% in terms of heterogeneous QoS provisioning when compared to FLS and RADS. However, this performance comes at the expense of the complexity increase as depicted in Fig. 5. Since the FP stage is performed for all traffic classes at almost each TTI, CACLA needs additional time resources in proportion of 20% to complete its tasks when compared to FLS, while the extra complexity requirement exceeds 30% when compared to RADS. Above this level, the complexity required by CACLA starts to stabilize or even to decrease since it behaves more like a SP scheme, while the FLS complexity becomes higher.

C. Practical Implications

According to our findings, some aspects must be considered when employing a RL-based scheduling solution for traffic prioritization, user scheduling and resource allocation in practice, such as: the training data set, the state space pre-processing, the controller configuration and termination condition for the learning stage. In order to get a generalised training data set, the training samples must consider variable number of users and changed at certain time intervals for each traffic class. Moreover, different speed levels and direction models should be considered for mobile users in order to explore a high variety of channel conditions. Under its original form, the training data-set is multidimensional and variable, depending on the number of active users that may change over time. Therefore, some pre-processing methods are necessary to compress the dimension of input state to some constant representations. Statistical methods can be used to get the mean and standard deviation values for the QoS indicators (i.e., packet loss, delay, throughput, etc.) for each traffic class [18]. Also, supervised learning can be used to classify the CQI reports in given patterns for users of each traffic class [31]. The optimal configuration of RL controller depends on the number of traffic classes and scheduling rules. When the number of traffic classes increases, higher number of hidden layers and nodes can be required with respect to some complexity constraints. Additionally, the output layer for the actor neural network must be properly managed and decoded in traffic class and scheduling rule selection as the size of the action space increases. During learning, both critic and actor errors must be monitored. In case of over-fitting (error increases above given threshold), the weights should be saved and learning process stopped. Otherwise, learning can continue for a number of iterations (TTIs) a priori established.

VI. Conclusion

This paper proposes an intelligent Machine Learning-based scheduling solution which makes use of Reinforcement Learning by employing CACLA, to react to the changeable networking conditions and take the best decisions in order to improve the fraction of time (in TTIs) when the QoS requirements are met for diverse services. Thus, the algorithm decides at each TTI the traffic class prioritization and the type of scheduling rule to be employed. Different traffic classes are dynamically prioritized such that the over-provisioning effect for some applications is avoided, whereas radio resources are intelligently managed by choosing the best scheduling rule for user scheduling and resource allocation. The proposed solution is deployed in a very challenging dynamic environment in which UAV performs UHD VR-based live video streaming to ground users. The proposed solution was evaluated through simulations and compared against other three state-of-the-art scheduling algorithms, such as: SP, RADS and FLS. The simulation results indicate that the proposed CACLA-based RL scheduling solution outperforms the other schemes involved while considering four perspectives: a) CACLA outperforms RADS and FLS in terms of packet loss, delay, throughput and PSNR when considering UHD VR-based users only; b) when considering a mixture of users requesting heterogeneous services, CACLA shows gains in excess of 100% by measuring the fraction of TTIs when the heterogeneous QoS requirements are respected; c) by measuring the inter-class packet loss, CACLA can accommodate a higher number of UHD VR users in the network, while SP and FLS prioritization schemes are over-provisioning some traffic classes; d) CACLA provides the best performance vs. complexity tradeoff.

Acknowledgment

G.-M. Muntean would like to acknowledge the Science Foundation Ireland grant 13/RC/2094 to Lero—the Irish Software Research Centre (http://www.lero.ie).

References

[1] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*, Cisco, San Jose, CA, USA, Feb. 2017. Accessed: Dec. 7, 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 133–144.

[3] L. Fay, L. Michael, D. Gómez-Barquero, N. Ammar, and M. W. Caldwell, "An overview of the ATSC 3.0 physical layer specification," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 159–171, Mar. 2016.

[4] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: Overview and Rel-11 enhancements," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 68–74, Nov. 2012.

[5] J. J. Gimenez *et al.*, "5G new radio for terrestrial broadcast: A forward-looking approach for NR-MBMS," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 356–368, Jun. 2019.

[6] X. Wang *et al.*, "Millimeter wave communication: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1616–1653, 3rd Quart., 2018.

[7] C. Ge *et al.*, "QoE-assured live streaming via satellite backhaul in 5G networks," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 381–391, Jun. 2019.

[8] A. Doumanoglou *et al.*, "A system architecture for live immersive 3D-media transcoding over 5G networks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 11–15.

[9] N. Jawad *et al.*, "Smart television services using NFV/SDN network management," *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 404–413, Jun. 2019.

[10] R. Viola, A. Martin, M. Zorrilla, and J. Montalbán, "MEC proxy for efficient cache and reliable multi-CDN video distribution," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 1–7.

[11] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[12] A. Martin *et al.*, "Network resource allocation system for QoE-aware delivery of media services in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 561–574, Jun. 2018.

[13] E. Mangina, E. O'Keeffe, J. Eyerman, and L. Goodman, "Drones for live streaming of visuals for people with limited mobility," in *Proc. 22nd Int. Conf. Virtual Syst. Multimedia (VSMM)*, Oct. 2016, pp. 1–6.

[14] A. Doumanoglou *et al.*, "Quality of experience for 3-D immersive media streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 379–391, Jun. 2018.

[15] R. Trestian, I.-S. Comşa, and M. F. Tuysuz, "Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?" *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 945–977, 2nd Quart., 2018.

[16] N. D. Tripathi and J. H. Reed. *5G Evolution: On the Path to 6G Expanding the Frontiers of Wireless Communications White Paper, Rohde & Schwarz Technical Document 2019*, Accessed: Nov. 6, 2019. [Online]. Available: https://www.mobilewirelesstesting.com/beyond-5g-intro-6g/

[17] M. Hosseini, Y. Jiang, R. R. Berlin, L. Sha, and H. Song, "Toward physiology-aware DASH: Bandwidth-compliant prioritized clinical multimedia communication in ambulances," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2307–2321, Oct. 2017.

[18] I.-S. Comşa, A. De-Domenico, and D. Ktenas, "QoS-driven scheduling in 5G radio access networks—A reinforcement learning approach," in *Proc. IEEE Global Commun. Conf. GLOBECOM*, Dec. 2017, pp. 1–7.

[19] S. Abedi, "Efficient radio resource management for wireless multimedia communications: A multidimensional QoS-based packet scheduler," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2811–2822, Nov. 2005.

[20] G. Piro, L. A. Grieco, G. Boggia, R. Fortuna, and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052–1065, Oct. 2011.

[21] G. Monghal, D. Laselva, P. Michaelsen, and J. Wigard, "Dynamic packet scheduling for traffic mixes of best effort and VoIP users in E-UTRAN downlink," in *Proc. IEEE 71st Veh. Technol. Conf.*, May 2010, pp. 1–5.

[22] X. Xie, X. Zhang, S. Kumar, and L. E. Li, "piStream: Physical layer informed adaptive video streaming over LTE," *GetMobile Mobile Comput. Commun.*, vol. 20, no. 2, pp. 31–34, Oct. 2016.

[23] X. Wang, A. Chowdhery, and M. Chiang, "Networked drone cameras for sports streaming," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 308–318.

[24] I.-S. Comşa *et al.*, "Towards 5G: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1661–1675, Aug. 2018.

[25] I.-S. Comşa, S. Zhang, M. Aydin, P. Kuonen, R. Trestian, and G. Ghinea, "A comparison of reinforcement learning algorithms in fairness-oriented OFDMA schedulers," *Information*, vol. 10, no. 10, p. 315, Oct. 2019.

[26] M. Elsayed and M. Erol-Kantarci, "AI-enabled radio resource allocation in 5G for URLLC and eMBB users," in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Nov. 2019, pp. 590–595.

[27] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 94–101, Feb. 2018.

[28] *Technical Specification Group Services and System Aspects; Policy and Charging Control Architecture Release 12, v.12.2.0*, 3GPP, Sophia Antipolis, France, 2013.

[29] G. Song and Y. Li, "Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 127–134, Dec. 2005.

[30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, U.K.: IMT Press, 2017.

[31] I.-S. Comşa, *Sustainable Scheduling Policies for Radio Access Networks Based on LTE Technology*, Ph.D. dissertation, Univ. Bedfordshire, Luton, U.K., 2014. AQ3

[32] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, R. Trestian, and G. Ghinea, "Guaranteeing user rates with reinforcement learning in 5G radio access networks," in *Next-Generation Wireless Networks Meet Advanced Machine Learning Applications*, vol. 8. Hershey, PA, USA: IGI Global, 2019, pp. 163–198.

[33] B. Sadiq, R. Madan, and A. Sampath, "Downlink scheduling for multi-class traffic in LTE," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, no. 14, pp. 1–18, 2009.

[34] N. Khan, M. Martini, Z. Bharucha, and G. Auer, "Opportunistic packet loss fair scheduling for delay-sensitive applications over LTE systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, vol. 1, Apr. 2012, pp. 1456–1461.

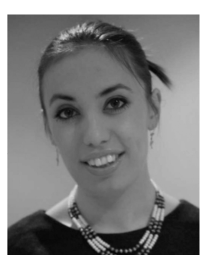
[35] H. Van Hasselt and M. Wiering, "Using continuous action spaces to solve discrete problems," in *Proc. Int. Joint Conf. Neural Netw.*, Apr. 2009, pp. 1149–1156.

[36] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498–513, Feb. 2011.

[37] A. Moldovan, I. Ghergulescu, and C. H. Muntean, "VQAMap: A novel mechanism for mapping objective video quality metrics to subjective MOS scale," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 610–627, Sep. 2016.

Ioan-Sorin Comşa received the B.Sc. and M.Sc. degrees in telecommunications from the Technical University of Cluj-Napoca, Romania, in 2008 and 2010, respectively, and the Ph.D. degree from the Institute for Research in Applicable Computing, University of Bedfordshire, U.K., in June 2015. He is a Research Scientist in 5G radio resource scheduling with Brunel University, London, U.K. He was a Ph.D. Researcher with the Institute of Complex Systems, University of Applied Sciences of Western Switzerland, Switzerland. Since 2015, he has been a Research Engineer with CEA-LETI, Grenoble, France. His research interests include intelligent radio resource and QoS management, reinforcement learning, data mining, distributed and parallel computing, and adaptive multimedia/mulsemedia delivery.

Gabriel-Miro Muntean (Senior Member, IEEE) is an Associate Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of the DCU Performance Engineering Laboratory. He has published over 350 papers in top-level international journals and conferences, authored four books and 18 book chapters, and edited seven additional books. His research interests include quality, performance, and energy issues related to rich media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and a reviewer for important international journals, conferences, and funding agencies. He is the Coordinator of the EU-funded project NEWTON http://www.newtonproject.eu.



Ramona Trestian (Member, IEEE) received the Ph.D. degree from Dublin City University, Ireland, in 2012. She is a Senior Lecturer with the Design Engineering and Mathematics Department, Middlesex University, London, U.K. She was with Dublin City University as an IBM/IRCSET Exascale Postdoctoral Researcher. She published in prestigious international conferences and journals and has five edited books. Her research interests include mobile and wireless communications, user perceived quality of experience, multimedia streaming, handover and network selection strategies, and digital twin modeling.