A Machine Learning Resource Allocation Solution to Improve Video Quality in Remote Education

Ioan-Sorin Comșa, Andreea Molnar, Irina Tal, Per Bergamin, Gabriel-Miro Muntean, *Senior Member, IEEE*, Cristina Hava Muntean and Ramona Trestian, *Member, IEEE*,

***Abstract*—The current global pandemic crisis has unquestionably disrupted the higher education sector, forcing educational institutions to rapidly embrace technology-enhanced learning. However, the COVID-19 containment measures that forced people to work or stay at home, have determined a significant increase in the Internet traffic that puts tremendous pressure on the underlying network infrastructure. This affects negatively content delivery and consequently user perceived quality, especially for video-based services. Focusing on this problem, this paper proposes a machine learning-based resource allocation solution that improves the quality of video services for increased number of viewers. The solution is deployed and tested in an educational context, demonstrating its benefit in terms of major quality of service parameters for various video content, in comparison with existing state of the art. Moreover, a discussion on how the technology is helping to mitigate the effects of massively increasing internet traffic on the video quality in an educational context is also presented.**

***Index Terms*—video quality, machine learning, resource allocation, quality of service, technology enhanced learning.**

I. Introduction

THE latest advancements in technologies have enabled the recent evolution of mobile device connectivity towards high-generation networks like 4G and 5G. The higher bandwidth availability and increased affordability and popularity of powerful high-end mobile devices have led to a wide spread and adoption of advanced multimedia applications on mobile devices. However, this in turn has determined a significant increase in mobile and wireless traffic that puts pressure on the underlying network connectivity. According to the predictions provided by Cisco [1] it is expected that by 2022 the number of Internet users will reach 60% of the global population, video traffic will account for 82% of all IP traffic, and the number of online connections and devices will surpass 28 billion.

However, these predictions might have been exceeded by reality due to the current global pandemic that forced most of the industries to rely on digital technologies. This global crisis revealed the importance of reliable and high speed mobile and wireless communication technologies and has opened up new directions around the issues of digital inclusiveness and connecting the unconnected [2]. The importance of broadband connectivity for mitigating the economic aftermath of the pandemic and boosting the digital access and inclusivity were aspects emphasized by the International Telecommunication Union (ITU) during an emergency virtual meeting of the Broadband Commission for Sustainable Development [3]. Another aspect, especially in the field of education, is digital equity, meaning that all regions of the world, not just the developed countries, can cope with it [4].

COVID-19 containment measures forced people to stay at home which fuelled the rapid digitalization of many sectors, including education. The actors in these sectors have started to deliver remotely large amounts of media content using diverse technologies across the existing network infrastructure.

Focusing on education, advanced multimedia applications delivered over current and next generation mobile networks, with improved support for seamless augmented reality (AR), virtual reality (VR) and mix reality (MR) are seen as key enablers for efficient e-learning [2]. These applications are efficient in a situated and practical learning context. However, to be able to accommodate good online learning for everyone and provide an appropriate level of e-learning for students and teachers, there is a need to go beyond the basic requirements of access to a digital device and stable internet connectivity. Quality in all its facets plays a fundamental role and supporting optimal experience for increasing number of users which exchange larger amounts of rich media content across the existing networks is of paramount importance [5].

This paper introduces Hierarchical Multi-Agent Reinforcement Learning (HiMARL), a novel machine learning solution to support optimized network resource allocation. HiMARL can accommodate a large number of learners consuming video content through mobile devices at acceptable quality within an educational wireless environment for remote education. In this context, this paper has the following contributions:

- it presents a subjective study involving diverse educational video content that analyses the impact of video quality levels on learning achievements. The results show that the video quality levels can be reduced with no significant impact on learning achievements, regardless of the video content.
- it proposes a dynamic scheduling solution that applies a controlled adjustment of video quality for learners

I.-S. Comșa and P. Bergamin are with the Institute for Research in Open-, Distance- and eLearning, Swiss Distance University of Applied Sciences, Brig, CH-3900, Switzerland (e-mails: ioan-sorin.comsa@ffhs.ch, per.bergamin@ffhs.ch).

P. Bergamin is also with the Faculty of Education, North-West University, Potchefstroom 2520, South Africa.

A. Molnar is with Swinburne University of Technology, Melbourne, Australia (e-mail: amolnar@swin.edu.au).

I. Tal and G.-M. Muntean are with Dublin City University, Dublin 9, Ireland (e-mails: irina.tal@dcu.ie, gabriel.muntean@dcu.ie). They acknowledge the support of Science Foundation Ireland grant 13/RC/2094_P2 to Lero.

C. H. Muntean is with National College of Ireland, Dublin 1, Ireland (e-mail: cristina.muntean@ncirl.ie)

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

requesting educational content simultaneously.
- it proposes HiMARL, a novel machine learning-based framework that increases the number of mobile learners that could be served over an Orthogonal Frequency Division Multiple Access-based (OFDMA) network with good quality video content and no impact on their learning experience.
- it deploys a prioritization policy that enables a dynamic prioritization of learners requesting heterogeneous video content from different classes to maximize the overall Quality of Service (QoS) provisioning.
- it introduces a novel hierarchical decision-making process based on multi-agent Reinforcement Learning (RL) that makes use of a master and a slave controller. The master controller is used to learn the most suitable prioritization sequence for diverse video classes. The slave controller is responsible to approximate the best scheduling rules that can be used for each video class. This novel approach enables scalability and flexibility of the overall proposed solution.

The rest of the paper is structured as follows. In Section II, the impact of increased internet traffic among others, driven by the consequences of the COVID-19 pandemic is discussed in terms of the perceived video quality and the resulting improved learning experience. Section III presents the results of the subjective tests focused on studying the effect of video quality variation. Section IV introduces the system model and optimization problem. The details of the HiMARL multi-agent RL framework are presented in Section V, whereas testing results and discussions are included in Section VI. Finally, Section VII concludes the paper.

II. Quality-oriented Video-based Learning

The rapid spread of COVID-19 has determined a forced mass migration to online delivery of content to remote users. Unlike any other sectors, in education, this migration has been almost total and has involved delivery of rich media content to remote users with highly heterogeneous operational environments, mostly in terms of device specifications and network characteristics. These aspects have severely affected content delivery and consequently, viewers' quality of experience and learners' level of education. To address this, of paramount importance is integration of innovative technology-based techniques and approaches in remote education [6].

In order to take into account the rapid transition to online learning, two approaches were chosen: synchronous and asynchronous learning, often integrated in a so-called blended learning approach. During synchronous learning sessions, students attend online live lectures and there is a real-time interaction between teachers and learners through various online platforms, such as Zoom, Google Classrooms, Kaltura Newrow, Microsoft Teams, Webex, etc. However, to be able to deliver a successful online learning experience through these online platforms, some aspects need to be taken into account: accommodate a high number of students to the video conferencing, easy two-way interaction between teachers and students, stable internet connectivity, availability of teachers and students, availability of teaching material, formative and summative feedback and assessment [7]. During asynchronous learning, students have access to prepared learning material, such as texts, pre-recorded video material, various forms of assessments, often provided through a learning management system (LMS) with the aim of either preparing or following up the synchronous phases. These two avenues made possible digital learning during the global pandemic period, although there are some quality-related concerns [8].

A. Quality of Learning Studies

Khattar et al. [9] studied the impact of the pandemic on the learning styles for students in India. The concern about their educational attainment motivated the students to adjust to an online-only learning method. However, the results show that almost half of the participants are facing broadband bandwidth and download speed limitations while only a few of the participants feel that the online classes provide them with appropriate structured learning activities during lockdown.

Mobile learning (m-learning) is also seen as a valuable solution for students and teachers in distance learning setups [10]. Apart from the fact that learning can occur at any-place, anywhere and at anytime, m-learning students benefit from developing their technological, conversational and high-ordered thinking skills [11]. The confinement circumstances caused by the global pandemic together with the necessity of continuing the education places m-learning as an essential educational technology component [12]. Romero-Rodriquez et al. [13] conducted a study to analyze the socio-demographic factors that influence m-learning during the pandemic in Spain. The study identified six such factors: teacher status, type of institution, educational technology, regular use of pedagogical innovations, use of mobile devices, belief in m-learning. However it noted that the lack of teacher technology training severely affects the adoption of m-learning. In this context, making use of the advancements in technologies like VR, AR, MR, 360-degree videos or images could enable the creation of a virtual learning environment that could prompt or guide efficient learning [14] and that could actually compensate for the lack of face-to-face interaction between teachers and learners. However, for a successful adoption and integration of m-learning while also supporting rich media content, there are several key factors that need to be considered: (1) availability of high specification devices; (2) availability of appropriate online materials; (3) good Internet connectivity and (4) technology-based solutions to bridge any gap which exists in terms of (1), (2) or (3). Next some solutions proposed in the context of (4) are discussed.

B. Quality-aware Multimedia Delivery Solutions

Recent research has focused on devising solutions for providing high user quality of experience (QoE) levels when delivering multimedia in variable network delivery conditions. Adaptive delivery solutions are among the most promising approaches proposed. Various multimedia adaptive delivery solutions have been described in the literature including server-located adaptive decision making solutions [15], client-located DASH-based approaches [16], solutions which focus on video adaptation only [17] or schemes which target rich media such

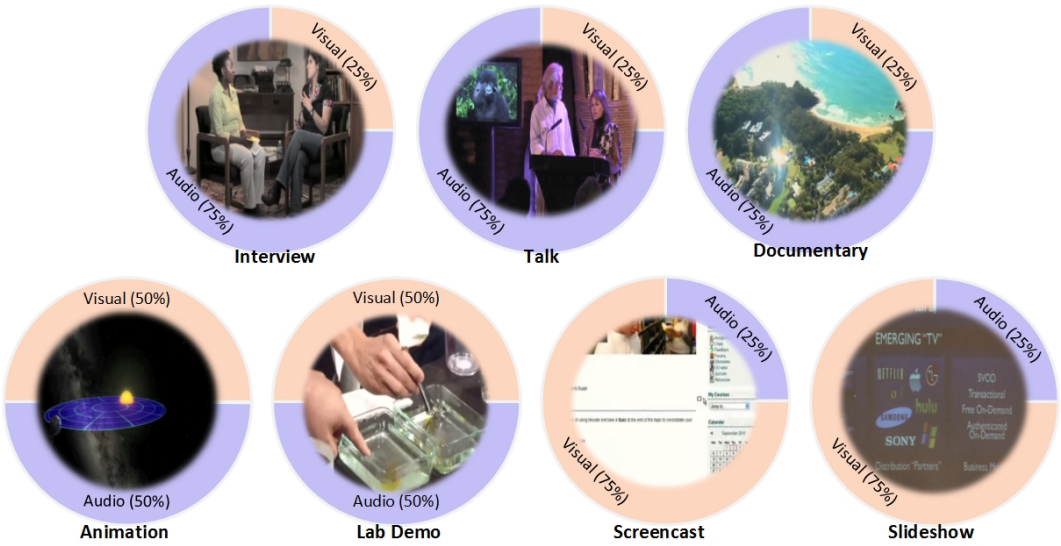


Fig. 1: Educational video content types used in the study

as multiple sensorial content delivery [18] or omnidirectional video [19] and finally generic adaptive solutions [20], device adaptation mechanisms [21] or network characteristics-aware adaptive schemes [22]. There are also commercial adaptive streaming solutions like Apple's HTTP Live Streaming (HLS), Microsoft's Silverlight and the more recent security-enhanced adaptive solution proposed by Akamai. Adaptive rich media delivery solutions help increase user QoE in general, and in the specific context of learning, research studies show that they also have potential to increase learner QoE and academic performance. However this is not simple and the proposed solutions have become increasingly complex. If the solutions also consider performing network resource allocation in a heterogeneous environment, there is a need for innovative approaches such as use of Machine Learning (ML). As an example, the ML-based OFDMA scheduling approach proposed in [23] can accommodate up to 50% more connections in terms of 360-degree and traditional video content when compared to state-of-the-art multi-class schedulers.

III. Video Quality Impact on Learning Achievements

This section presents a study on the impact of educational video content and video quality level on learning achievements of mobile learners [24].

A. Educational Video Content

The cognitive theory of multimedia learning [25] suggests that one can achieve better learning by structuring the multimedia materials effectively. The study shows that the capacity to remember can be increased and the learning experience can be maximized by presenting the educational content both audio and visually.

Seven educational content video clips belonging to seven categories were selected [26], as illustrated in Fig. 1 and defined as: *(1) Interview* - one or more participants answers questions; *(2) Talk/Presentation* - the speaker and the accompanying slides are presented; *(3) Documentary* - higher number of scene changes with outdoor locations; *(4) Animation* - computer generated clip of virtual world; *(5) Lab Demo* - a person demonstrates how to do a certain practical thing; *(6) Screencast* - recordings of the computer screen; and *(7) Slideshow* - recordings of voice over slides. Figure 1 also presents an estimation of the amount of educational information spread across audio and visual components for each video sequence. The educational content selected meets the requirements of various audiences from different domains, and avoids the potential situation that participants are interested in particular media types.

B. Subjective Tests

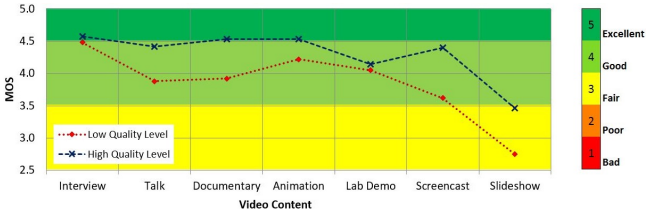
The aim of this subjective study is to understand if the mobile learners could achieve the desired learning experience across different video content categories, regardless of the video quality. A total of 54 participants (72% male, 28% female) aged from 19 to 57 have participated in the study. Each educational video content type was encoded at two different quality levels (e.g., bit rate): low quality level (i.e., 600kbps) and high quality level (i.e., 1Mbps). Each participant watched all the video sequences at both quality levels in a randomized order. Standard recommendations for subjective assessment of video quality were followed as described in [27]. After watching each video, the participants rated the overall quality level using the classic 5-point scale: 1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent and following the Absolute Category Ranking (ACR) method. Apart from assessing the user perceived quality of the educational content, the subjects were also given pre- and post-video watching questionnaires related to the video content, in order to measure their learning achievements.

C. User Perceived Educational Video Quality Results

The users were instructed to rate the video quality using the Mean Opinion Score (MOS) scale only and not the educational content as such. The MOS results for the seven categories are presented in Fig. 2. On average, across all seven categories both low and high quality levels were perceived as *Good* with MOS of 3.85 and 4.29, respectively. Considering the category

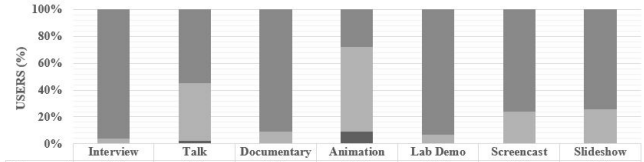
TABLE I: **Student t-test Results**

	Talk	Documentary	Animation	Screencast	Slideshow	Interview	Lab Demo
t-value	-6.00	-4.79	-3.26	-6.84	-5.29	-1.30	-0.56
p-value	<0.01	<0.01	<0.01	<0.01	<0.01	0.19	0.57



Video Content	Interview	Talk	Documentary	Animation	Lab Demo	Screencast	Slideshow
Low Quality Level	4.48	3.88	3.92	4.22	4.05	3.62	2.75
High Quality Level	4.57	4.42	4.53	4.53	4.14	4.40	3.46

Fig. 2: MOS results



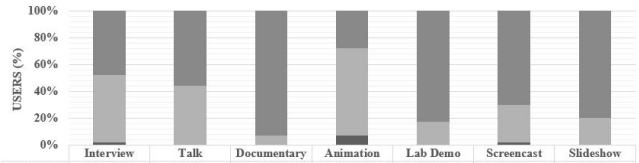
	Interview	Talk	Documentary	Animation	Lab Demo	Screencast	Slideshow
Improved	96%	55%	91%	28%	93%	76%	74%
Same	4%	43%	9%	63%	7%	24%	26%
Declined	0%	2%	0%	9%	0%	0%	0%

Fig. 3: Learning achievement for low quality educational video

type, the low quality level was perceived as *Good* across all seven categories apart from *Slideshow* which was perceived as *Fair*. The high quality level was perceived as *Excellent* for *Interview, Documentary and Animation*, as *Good* for *Talk, Lab Demo and Screencast* and *Fair* for *Slideshow*. It was noted that the video clips that contain text (e.g., *Screencast* and *Slideshow*) were rated slightly lower than the rest of the categories. In order to perform statistical analysis, t-tests were used [28] and the results are listed in Table I. While five out of seven categories (i.e., *Talk*, *Documentary*, *Animation*, *Screencast*, and *Slideshow*) show a statistically significant difference between the ratings obtained for the two quality levels, for the other two categories (i.e., *Interview*, *Lab Demo*) no statistically significant differences were observed.

D. Learning Achievements Results

In order to determine the learning achievements, we compared the pre- and post- questionnaire answers and defined three levels of learning achievements: (1) *Improved* - wrong answer to pre- questionnaire and correct answer to post-questionnaire, (2) *Stationary* - the same answer to pre- and post- questionnaires, and (3) *Declined* - correct answer to pre-questionnaire and wrong answer to post- questionnaire. We computed the percentage of users for each learning achievement level and for each educational content type as illustrated in Fig. 3 and 4 for low quality and high quality levels, respectively. We found that a statistical significant difference is obtained regardless of the video quality level or video content type with a confidence interval of 95%. The results show that most of the learners have an improved learning achievement or they maintain it at the same level, regardless of the video quality level. This study shows that the students can learn regardless of the educational content category or video quality used (i.e., low or high quality levels).



	Interview	Talk	Documentary	Animation	Lab Demo	Screencast	Slideshow
Improved	48%	56%	93%	28%	83%	70%	80%
Same	50%	44%	7%	65%	17%	28%	20%
Declined	2%	0%	0%	7%	0%	2%	0%

Fig. 4: Learning achievement for high quality educational video

IV. System Model

Based on the subjective studies previously conducted, this section presents the system model, optimization problem and the proposed solution when employing OFDMA as a radio access technology. The choice of the OFDMA access scheme is motivated by: *a)* simplicity and efficiency when compared to non-orthogonal access technologies, and *b)*, the proposed framework is intended for educational systems in developing countries where the beyond 4G communications systems are not yet employed. The proposed model could be integrated in more advanced radio access networks where the scheduling and resource allocation must be adapted accordingly and the HiMARL framework retrained.

The proposed system, illustrated in Fig. 5, is composed of a mobile learning server that stores the educational content to be streamed and an intelligent scheduler located at the level of the Base Station (BS). We assume that there is no communication loss on the wired link between the server and BS. Multiple mobile learners are connected to the same BS by using the OFDMA radio access technology. All mobile learners have different requests in terms of the educational video content that is downloaded simultaneously. The role of the network scheduler is to share the radio resources among multiple mobile learners to assure acceptable QoS for the delivered educational content. However, an acceptable QoS is challenging in OFDMA communications due to the variability of wireless channels in both time and frequency domains. Depending on the learner profile and device characteristics, the type (i.e interview, talk, animation) and load of educational video traffic can differ over time from one learner to another. Moreover, given the pricing schemes imposed by the network operator, a variety of video traffic characteristics are involved since a lower video quality is preferred by those learners who are not necessarily willing or cannot afford to pay too much for this service, while high video quality can be requested by other mobile learners with much better financial situation [29]. Given the multitude of characteristics and requirements, an intelligent network scheduler is needed in technology-enhanced learning systems to accommodate a higher number of mobile learners without degrading their learning experience.

The educational video content is organized in different traffic classes that depend on video quality and encoding bit rates, where each class is characterized by a specific set of

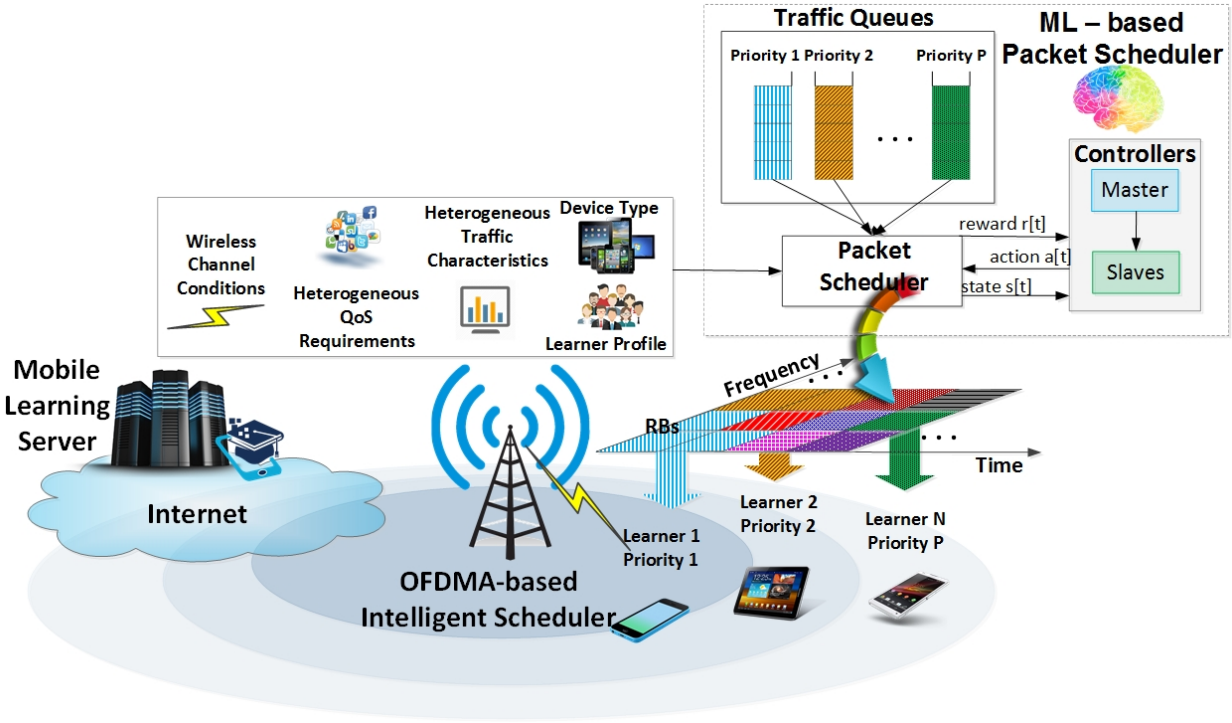


Fig. 5: Proposed ML-based framework for mobile learners

QoS requirements in terms of throughput, packet loss and latency. Given the COVID-19 pandemic situation, the network operator needs to prioritize the disadvantaged mobile learners requesting the educational video content at lower bit rates compared to other learners being served at much higher video bit rates. As seen in Fig. 5, at the scheduler level, data packets belonging to video traffic class with the lowest bit rates are waiting to be transmitted in data queues with priority 1 while packets corresponding to learners with the highest bit rates are awaiting in data queues with priority P, where P is the total number of video classes. In general, the OFDMA packet scheduler is designed to work in two stages [23]: *a)* Time Domain Prioritization (TDP) where learners with more stringent QoS requirements are prioritized over other learners requesting video content with more relaxed QoS budget; and *b)* the Frequency Domain Prioritization (FDP), where the pre-selected group of learners are competing in getting the best radio resources to receive the requested video services. Both stages are iteratively performed at each Transmission Time Interval (TTI) to respect the QoS requirements for all P video classes. Since the networking conditions (variability of wireless channels, learner profiles, traffic characteristics and QoS requirements) are changing at each TTI, most of the existing packet schedulers are unable to adapt at the newest conditions leading to over-provisioning of some classes (i.e. most prioritized learners with the lowest video bitrate) while degrading the QoS provisioning of other learners requesting higher bit rates traffic. In frequency domain, the most used FDP schemes are rather static being focused on a particular QoS target leading to unbalanced QoS multi-objective optimization [30]. In this paper, we propose a dynamic network scheduler able to adapt the prioritization in both time and frequency domains in order to improve the QoS provisioning of all learner classes. The proposed solution makes use of Machine Learning (ML) to enhance decision-making for all traffic classes and in different networking conditions. The aim is to increase the number of learners that can access heterogeneous education based on video with insignificant decrease of their learning experience.

The proposed ML architecture employs an interaction at each TTI between an intelligent controller and the network scheduler as depicted in Fig. 5. The controller is designed to learn the most appropriated scheduling actions based on the current states. The scheduling decisions to be considered at each TTI controls both TDP and FDP stages. In this sense, the proposed controller employs a hierarchical ML architecture organized as follows: *a)* a master controller that adapts the TDP stage by deciding at each TTI, the order of video classes to be prioritized in the frequency domain; *b)* a slave controller that decides the scheduling rule for the resource prioritization in the FDP domain for each video class. The role of the master controller is to avoid the over-provisioning effect of some video classes (low bit-rates, high priority) while helping other classes with higher video bit rates to meet the corresponding QoS requirements. The slave controller aims at meeting the multi-objective target given the multitude of QoS requirements in terms of packet loss, throughput and latency for each video traffic class. The objective of the proposed hierarchical ML-based framework is to accommodate higher number of learners in the network from all classes of video traffic while keeping good level of learning experience.

A. OFDMA Scheduling Model

Given a prioritization order, we define by $\mathcal{P} = \{1, 2, ..., P\}$ the set of video classes, where class 1 represents the video traffic with the highest priority to be delivered while class P

TABLE II: **List of Notations**

Parameter	Description
$\mathcal{A}$	Controller action space
$\mathbf{a}$	Controller action $\mathbf{a} \in \mathcal{A}$ decided at TTI t
$\mathcal{A}_M$	Master controller action space
$\mathbf{c}$	Master controller action $\mathbf{c} \in \mathcal{A}_M$ decided at TTI t
$\mathcal{A}_S$	Slave controller action space
$\mathbf{r}$	Slave controller action $\mathbf{r} \in \mathcal{A}_S$ decided at TTI t
$\mathcal{B}$	Set of resource blocks
b	Random resource block $b \in \mathcal{B}$
B	Max. no. of resource blocks
δ_p	Time Difference error of slave agent $p \in \mathcal{P}$
Δ_p	Time Difference error of master agent $p \in \mathcal{P}$
η	Learning rate
γ	Discount factor
$\mathcal{O}$	Set of heterogeneous objectives
$\mathcal{O}_p$	Set of objectives corresponding to class $p \in \mathcal{P}$
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 video classes in the priority order given by [31]
p	Random video class $p \in \mathcal{P}$
P	Max. no. of video classes
$\mathcal{P}^*(t)$	Set of scheduled video classes at TTI t
p^*	Random video class $p^* \in \mathcal{P}^*(t)$
P^*	Max. no. of video classes scheduled at TTI t
$\mathcal{P}^{\otimes}(t)$	Set of unscheduled video classes at TTI t
$p^{\otimes}$	Random video class $p^{\otimes} \in \mathcal{P}^{\otimes}(t)$
$P^{\otimes}$	Max. no. of video classes unscheduled at TTI t
$\mathcal{R}$	Set of scheduling rules
r_p	Random scheduling rule $r_p \in \mathcal{R}$ for class $p \in \mathcal{P}$
R	Max. no. of scheduling rules from $\mathcal{R}$
π_p	Policy value of slave agent $p \in \mathcal{P}$
Π_p	Policy value of master agent $p \in \mathcal{P}$
Q_p	Action-Value Function of slave agent $p \in \mathcal{P}$
V_p	Value Function of slave agent $p \in \mathcal{P}$
W_p	Action-Value Function of master agent $p \in \mathcal{P}$
V	Value Function of the entire controller state $\mathbf{s} \in \mathcal{S}$
$\mathcal{S}$	Continuous and multi-dimensional scheduler state space
$\mathbf{s}$	Momentary scheduler state $\mathbf{s} \in \mathcal{S}$ at TTI t
$\mathcal{S}_p$	Continuous and multi-dimensional state space of class $p \in \mathcal{P}$
$\mathbf{s}_p$	Momentary scheduler state $\mathbf{s}_p \in \mathcal{S}_p$ at TTI t
$\mathcal{L}$	Set of heterogeneous learners
L	Number of heterogeneous learners
$\mathcal{L}_p$	Set of learners corresponding to class $p \in \mathcal{P}$
L_p	Number of learners in class $p \in \mathcal{P}$
l	Learner index belonging to a given class $\mathcal{L}_p$
$x_{p,l,o}$	KPI indicator of $o \in \mathcal{O}_p$ and learner $l \in \mathcal{L}_p$
$\bar{x}_{p,l,o}$	KPI requirement of $o \in \mathcal{O}_p$ and learner $l \in \mathcal{L}_p$
t	TTI index
θ_p	Set of neural network weights for slave agent $p \in \mathcal{P}$
Θ_p	Set of neural network weights for master agent $p \in \mathcal{P}$
$\rho(\mathbf{s}, \mathbf{a})$	System reward when applying action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{s} \in \mathcal{S}$
ϱ_p	Reward value of slave agent $p \in \mathcal{P}$

is assigned with the lowest priority over P number of classes. A mobile learner is characterized by a mobile equipment (i.e. tablet, smartphone or laptop with broadband connectivity) and can request educational video content belonging to one of the class $p \in \mathcal{P}$. In this sense, we define by $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_P\}$ the set of all mobile learners requesting heterogeneous video content with different QoS requirements, where $\mathcal{L}_p$ is a subset of mobile learners requesting educational video content from class $p \in \mathcal{P}$. To achieve the desired learning experience, the delivery of the requested video content must respect the QoS requirements associated with traffic class $p \in \mathcal{P}$. The list of notations used in this paper is given in Table II.

Now, we define an objective as the intent of respecting a given QoS requirement for a specific video service requested by a given mobile learner. By $\mathcal{O}_p$ we mean the set of all objectives associated to class $p \in \mathcal{P}$ in terms of meeting the corresponding QoS requirements. Precisely, objective $o \in \mathcal{O}_p$ is met for the provided video service of leaner $l \in \mathcal{L}_p$, if the corresponding online Key Performance Indicator (KPI) $x_{p,l,o}$ respects its QoS requirement $\bar{x}_{p,l,o}$. Extending a little bit this concept in the direction of the multi-objective optimization of mobile learner $l \in \mathcal{L}_p$ requesting the video traffic $p \in \mathcal{P}$, we consider by $\mathbf{x}_{p,l} = [x_{p,l,o_1}, ..., x_{p,l,o_{O_p}}]$ the vector of KPIs of all objectives $\mathcal{O}_p$ and by $\bar{\mathbf{x}}_{p,l} = [\bar{x}_{p,l,o_1}, ..., \bar{x}_{p,l,o_{O_p}}]$ the associated requirement vector, where O_p is the number of QoS objectives of class $p \in \mathcal{P}$. Thus, learner $l \in \mathcal{L}_p$ meets all objectives $\mathcal{O}_p$ if $\mathbf{x}_{p,l}$ respects $\bar{\mathbf{x}}_{p,l}$. When considering the multi-objective optimization at the level of all mobile learners of class $p \in \mathcal{P}$, then the vector $\mathbf{x}_p = [\mathbf{x}_{p,l_1}, \mathbf{x}_{p,l_2}, ..., \mathbf{x}_{p,l_{L_p}}]$ represents the online KPI vector of class $p \in \mathcal{P}$ and $\bar{\mathbf{x}}_p = [\bar{\mathbf{x}}_{p,l_1}, \bar{\mathbf{x}}_{p,l_2}, ..., \bar{\mathbf{x}}_{p,l_{L_p}}]$ is the corresponding requirement vector, where L_p is the number of mobile learners receiving video content from class $p \in \mathcal{P}$. In this case, the entire set of mobile learners $\mathcal{L}_p$ meets all QoS objectives $\mathcal{O}_p$ if $\mathbf{x}_p$ respects $\bar{\mathbf{x}}_p$. When considering both multi-objective and multi-class optimization problem, we store in $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_P]$ the online KPIs of all mobile learners and we keep the set of QoS requirements in $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_P]$. To this extent, the entire set of objectives $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_P\}$ is met in all video classes $\mathcal{P}$, if $\mathbf{x}$ respects its requirement vector $\bar{\mathbf{x}}$. The aim of the proposed hierarchical-ML approach is to solve the multi-objective and multi-class optimization problem at each TTI such that the KPI vector $\mathbf{x}$ meets the requirement $\bar{\mathbf{x}}$ in the highest possible measure.

According to the employed scheduling strategies in both TDP and FDP domains, the KPI vector $\mathbf{x}$ is adjusted to respect its requirement $\bar{\mathbf{x}}$ in a certain measure given the wireless and network conditions. However, most of the schedulers aims to split the multi-objective optimization between the TDP and FDP stages. As a matter of this fact, in [32]–[34] users are prioritized in time domain based on the throughput, delay and packet loss indicators, while the frequency prioritization addresses the trade-off between system capacity maximization and meeting certain fairness criteria. Or some other schedulers are oriented more on minimizing the throughput loss caused by the time domain prioritization and employ heuristic-based resource allocation methods [35], [36]. From the perspective of multi-class scheduling, most of the existing strategies aims to prioritize a mixture of users from different classes for which the QoS requirements are not met without any precise consideration of prioritization order given by the sequence $[1, 2, ..., P]$. As a consequence, some lower prioritized classes can get overall more resources than that of other classes with higher priority level. Moreover, most of the existing TDP-FDP strategies are unable to react to the changeable networking conditions, leading to the over-provisioning of some traffic classes. However, to deal with these challenges, in [5] a Reinforcement Learning(RL)-based solution is proposed to dynamically prioritize one traffic class (i.e. $[p, 1, ..., P]$, $\forall p \in \mathcal{P}$) each time among others. Given the COVID-19 pandemic situation and the video content diversity, this solution may limit over time the mobile learners to get the requested services with enhanced level of QoS provisioning.

Once the mobile learners are prioritized, in frequency domain the available system bandwidth is divided into B number of equally distributed Resource Blocks (RBs), where each RB is the smallest radio resource that can be allocated. We define by $\mathcal{B} = \{1, 2, ..., B\}$ the set of RBs to be allocated by the FDP scheduler at each TTI, where B is the maximum

number of RBs. The allocation of $\mathcal{B}$ is conducted according to some scheduling rules that are considered various and oriented on particular QoS objective(s) [37]–[39]. Given this variety, let $\mathcal{R} = \{1, 2, ..., R\}$ be the set of scheduling rules with R elements. The problem is that once selected a scheduling rule $r \in \mathcal{R}$, the multi-objective optimization can be unbalanced in the frequency domain since only a particular QoS objective is addressed. Alongside of the traffic class prioritization, in [5] a particular scheduling rule $r \in \mathcal{R}$ is decided to be applied at each TTI for all traffic classes. However, since the needs in terms of QoS provisioning may differ from one traffic class to another, then a separate scheduling rule may be required. In this sense, a separate decision-making entity would be necessary to select an appropriate scheduling rule $r_p \in \mathcal{R}$ for each video class $p \in \mathcal{P}$.

This work proposes a hierarchical RL-based framework able to control the decisions for both TDP and FDP schedulers with the goal of accommodating a higher number of mobile learners requesting video content from all classes while keeping the desired level of the learning experience. The proposed architecture makes use of two controllers: *a)* the master controller decides based on the current network conditions the best prioritization sequence (i.e. $[p, P, 1, ..., p+1, p-1]$) at each TTI; *b)* the slave controller selects a particular scheduling rule $r_p \in \mathcal{R}$ for each class from the prioritization sequence according to the available bandwidth. Compared to [5], this approach can offer better scalability options and enhanced heterogeneous QoS provisioning.

B. Multi-Class and Multi-Objective Optimization

In the following, we formulate the aggregate multi-objective and multi-class optimization problem addressed at each TTI t, such as:

$$\max_{m,n,i} \sum_{r\in\mathcal{R}} \sum_{p\in\mathcal{P}} \sum_{l\in\mathcal{L}_p} \sum_{b\in\mathcal{B}} m_{r,p}(t) \cdot n_{p,l}(t) \cdot i_{l,b}(t) \cdot \Gamma_{r,p}[\mathbf{x}_{p,l}(t)] \cdot \lambda_{l,b}(t), \tag{1}$$

s.t.

$$\sum\nolimits_l i_{l,b}(t) \leq 1, \quad b = 1, ..., B, \tag{1.a}$$

$$\sum\nolimits_p n_{p,l}(t) = 1, \quad l = l_1, ..., l_{L_p}, p = 1, ..., P, \tag{1.b}$$

$$\sum\nolimits_{p^*} \sum\nolimits_l n_{p^*,l}(t) = \sum\nolimits_{p^*} L_{p^*}, \quad p^* \in \mathcal{P}^*, \mathcal{P}^* \subseteq \mathcal{P}, \tag{1.c}$$

$$\sum\nolimits_{p^\otimes} \sum\nolimits_l n_{p^\otimes,l}(t) = 0, \quad \forall p^\otimes \in \mathcal{P}^\otimes, \mathcal{P}^\otimes = \mathcal{P} \backslash \mathcal{P}^*, \tag{1.d}$$

$$\sum\nolimits_r m_{r,p^*}(t) = 1, \quad p^* \in \mathcal{P}^*, \tag{1.e}$$

$$\sum\nolimits_r m_{r,p^\otimes}(t) = 0, \quad p^\otimes \in \mathcal{P}^\otimes, \tag{1.f}$$

$$m_{r,p}(t) \in \{0, 1\}, \quad \forall r \in \mathcal{R}, \forall p \in \mathcal{P}, \tag{1.g}$$

$$n_{p,l}(t) \in \{0, 1\}, \quad \forall p \in \mathcal{P}, \forall l \in \mathcal{L}_p, \tag{1.h}$$

$$i_{l,b}(t) \in \{0, 1\}, \quad \forall l \in \mathcal{L}_p, \forall b \in \mathcal{B}. \tag{1.i}$$

where t is the TTI index from a given period of the optimization time, $m_{r,p} \in \{0, 1\}$ assigns a scheduling rule to each learner class (i.e. $m_{r,p} = 1$ if $r \in \mathcal{R}$ is assigned to $p \in \mathcal{P}$, and $m_{r,p} = 0$, otherwise), $n_{p,l} \in \{0, 1\}$ preselects learners for the frequency allocation (i.e. $n_{p,l} = 1$ if learner $l \in \mathcal{L}_p$ is preselected, and $n_{p,l} = 0$, otherwise) and $i_{l,b} \in \{0, 1\}$ allocates the preselected learners to get the available RBs (i.e. $i_{l,b} = 1$ if RB $b \in \mathcal{B}$ is allocated to learner $l \in \mathcal{L}_p$, and $i_{l,b} = 0$, otherwise). Variable $n_{p,l}$ is related more to the TDP scheduler while $m_{r,p}$ and $i_{l,b}$ are the variables that are determined at each TTI t by the FDP scheduler.

In (1), each scheduling rule $r \in \mathcal{R}$ of each video class is associated with the utility function $\Gamma_{r,p} : \mathbb{R} \to \mathbb{R}$, that takes as input the KPI indicator $\mathbf{x}_{p,l}$ and gives as an output value the priority of allocating learner $l \in \mathcal{L}_p$ in the frequency domain [40]. However, due to the changeable wireless channels in the frequency domain, the prioritization decision must be determined RB-by-RB, by calculating the following metric: $\Gamma_{r,p}(\mathbf{x}_{p,l}) \cdot \lambda_{l,b}$, $\forall b \in \mathcal{B}$, $\forall l \in \mathcal{L}_p$, $\forall r \in \mathcal{R}$ [40]. Here, parameter $\lambda_{l,b}$ is the achievable rate that would be obtained if RB $b \in \mathcal{B}$ is allocated with learner $l \in \mathcal{L}_p$. The calculation of the achievable rate at each TTI depends on the Channel Quality Indicator (CQI) vector that is reported by each learner $l \in \mathcal{L}_p$ to the serving BS. The solution to (1) aims to find at each TTI the best preselection decision of learners $l \in \mathcal{L}_p$ in time domain and the best scheduling rule $r_p \in \mathcal{R}$ to be followed to optimally allocate learners in the frequency domain. The target is to increase as much as possible at each TTI the learner benefit when the KPI vector $\mathbf{x}$ respects its requirement $\bar{\mathbf{x}}$.

Compared to other TDP strategies where users from different classes may be pre-selected according to their QoS budget [32], [33], the proposed scheduler aims to prioritize classes by deciding at each TTI a new prioritization sequence. However, the order of classes to be scheduled at each TTI is decided based on the performance of $\mathbf{x}$ over the requirement $\bar{\mathbf{x}}$ and not on the occupancy degree of the available spectrum. As a consequence, depending on the traffic load, some classes may remain unscheduled since all RBs are allocated to learners from classes with higher priorities decided at TTI t. In this sense, let us consider the set of video classes being composed as follows: $\mathcal{P} = \{\mathcal{P}^*(t), \mathcal{P}^\otimes(t)\}$, where $\mathcal{P}^*(t)$ is the set of scheduled classes while $\mathcal{P}^\otimes(t)$ are the classes that remain unscheduled at TTI t. When $\mathcal{P}^\otimes(t) = \{\varnothing\}$, then all classes $\mathcal{P}$ are allocated in frequency given the prioritization sequence.

Solving the multi-class and multi-objective optimization problem in (1) involves a set of constraints that must be respected. For example, constraints (1.a) indicate that one RB is allocated to at most one preselected learner $l \in \mathcal{L}_p$. To simplify the optimization model, in (1.b) each learner is constrained to request one traffic class with a given video quality at once. Only a number of L_{p^*} learners with $p^* \in \mathcal{P}^*$ is preselected to get allocated in the frequency domain as indicated by constraints (1.c), while learners from classes $p^\otimes \in \mathcal{P}^\otimes$ remain unscheduled as constrained by (1.d). A separate scheduling rule $r_{p^*} \in \mathcal{R}$ is decided for each class $p^* \in \mathcal{P}^*$ as shown in (1.e). However, to reduce the computational complexity, the selection of scheduling rules for classes $p^\otimes \in \mathcal{P}^\otimes$ that remain un-allocated at TTI t should be deactivated, as indicated in constraint (1.f). Finally, constraints (1.g)-(1.i) make the proposed optimization problem combinatorial.

In real practice, finding optimal solutions in (1) is difficult to achieve due to the very high searching space and time constraints. Also, the performance of applying a certain class

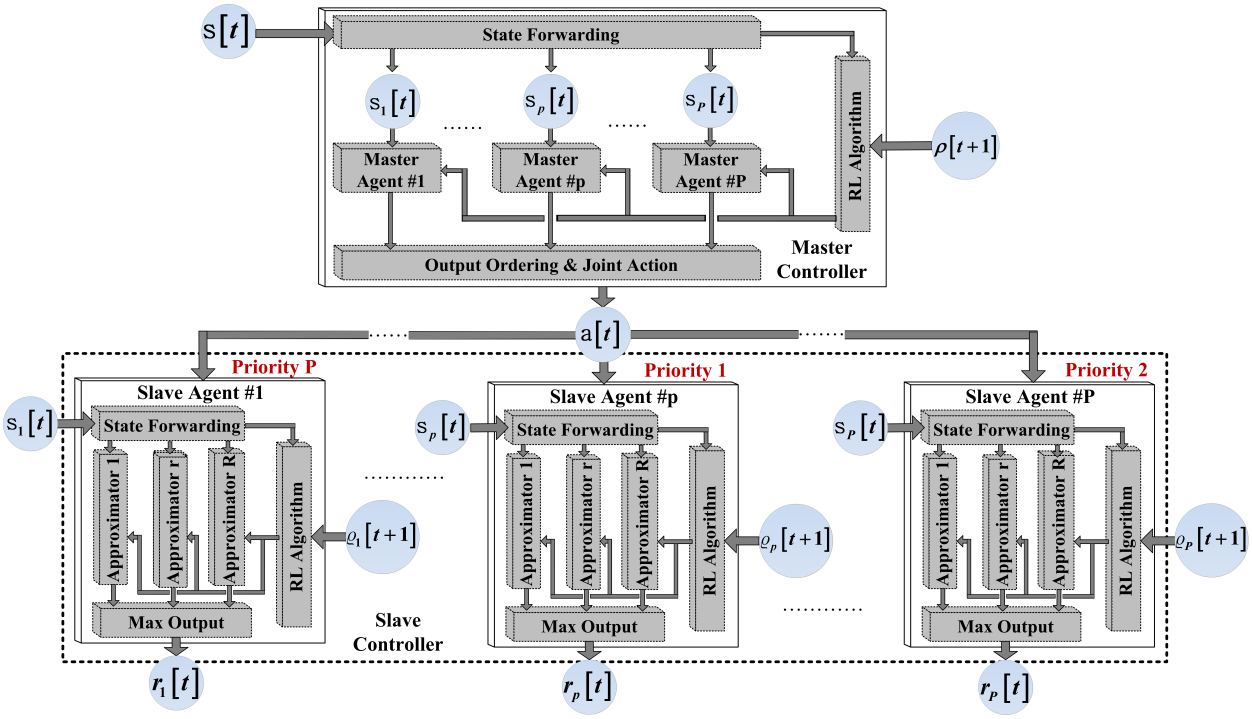


Fig. 6: Internal functionality of the proposed controller

prioritization sequence with the associated scheduling rules can be measured once the system evolves to the next state at TTI $t+1$. To this extent, the initial optimization problem is divided into three smaller sub-problems: *a)* the first sub-problem aims to find based on the actual networking conditions the most suitable prioritization sequence of video classes as part of the TDP process; *b)* the second sub-problem aims to find a proper scheduling rule $r_{p^*} \in \mathcal{R}$ for each class $p^* \in \mathcal{P}^*$; *c)* the third sub-problem performs the resource allocation of $\mathcal{B}$ given the priority order decided in *a)* and the scheduling rules obtained with *b)*. The proposed framework aims to intelligently allocate the radio spectrum at each TTI while providing solutions to *a)* and *b)* in order to maximize the QoS provisioning in all video classes.

C. Proposed Solution

The proposed controller from Fig. 5 is employed to find good solutions at each TTI in terms of the prioritization scheme and scheduling rules to be used for each video class. Based on the scheduler-controller interaction, these decisions can be learnt over time to maximize as much as possible the QoS provisioning for all traffic classes. However, due to the high dimension of the scheduler state space, these pairs cannot be exhaustively enumerated and hence, the optimal decisions can be only approximated. Thus, the aim is to learn the parameterization of some non-linear functions to approximate the best decisions afferent to sub-problems *a)* and *b)*.

To increase the scalability and flexibility of our solution, the internal structure of the proposed controller employs a hierarchical decision-making process as shown in Fig. 6. At the first level, the master controller learns to approximate the best prioritization sequence at each TTI. A number of P master agents are separately trained to claim the priority value of each video class. A separate entity is needed to order the agents' decisions and to form a joint action. At the second level, the slave controller is responsible to approximate the best scheduling rules that can be used for each video class from the prioritization sequence by training a group of P slave agents. Given the characteristics' diversity of different traffic classes, each Slave Agent (SA) is associated with a particular class and learns R number of non-linear functions to approximate the best selections of scheduling rules. In the learning stage, both master and slave agents are learnt to approximate the best decisions based on the RL algorithms. In the exploitation stage, the hierarchical structure applies the learnt decisions in terms of the class prioritization and the type of scheduling rule to be employed in the frequency domain in each state.

V. The Proposed HiMARL Framework

The scheduler-controller interaction as depicted in Fig. 5 is modeled based on: *states*, *actions* and *rewards*. At each TTI t, the controller observes the current state, takes an action and the scheduling process is conducted accordingly. At TTI $t+1$, a new state is observed and the rewards are computed to measure the performance of the applied action in the previous state. The proposed controller aims to explore high number of state-action-reward-state experiences and to learn over time to approximate the best actions to be applied in each state.

Given the hierarchical decision-making scheme in Fig. 6, the states, actions and rewards must be identified for each controller. The master and slave agents consider as states the current observations associated with the video class. At the master level, agents provide a joint action in terms of the prioritization sequence. At the slave level, agents provide the scheduling rule to be employed by each particular video class when the resource allocation is performed in the frequency domain. The rewarding scheme is organized as follows: *a)* the master controller receives a reward that quantifies the impact

of applying the prioritization sequence in the previous state; *b)* for the slave controller, the reward of each agent should quantify the impact of the applied scheduling rule from the perspective of the multi-objective QoS. These rewards are obtained only by those agents that represent the scheduled classes $p^* \in \mathcal{P}^*(t)$. Since all agents learns on different states and reward functions, then the master and slave controllers work under the Multi-Agent RL (MARL) regime.

The complexity of HiMARL framework depends on how many video classes and scheduling rules are used since a number of $P(R+1)$ function approximators must be trained. By training the slave and master controllers concomitantly, the impact of the applied actions cannot be differentiated. As an example, when scheduling video class $p^* \in \mathcal{P}^*(t)$, the reward amount afferent to the slave agent is influenced by the prioritization decision. To this extent, we train the framework in two steps: *a)* train first the slave controller and *b)* train the master controller while exploiting the slave decisions.

A. State Space

We define by $\mathcal{S}_p = \mathcal{S}_p^U \bigcup \mathcal{S}_p^C$ the measurable, multi-dimensional and continuous state space of video class $p \in \mathcal{P}$, where $\mathcal{S}_p^U$ and $\mathcal{S}_p^C$ are the uncontrollable and controllable sub-spaces, respectively. On one side, the uncontrollable sub-space $\mathcal{S}_p^U$ contains some stochastic elements such as CQI reports, number of active learners per video class, etc. On the other side, data points in $\mathcal{S}_p^C$ evolves as the result of the scheduling decisions each TTI. If $\mathbf{v}_p(t) \in \mathcal{S}_p^U$ and $\mathbf{y}_p(t) \in \mathcal{S}_p^C$ are the uncontrollable and controllable sub-states, respectively, at TTI t, then we define by $\mathbf{s}_p(t) = [\mathbf{v}_p, \mathbf{y}_p] \in \mathcal{S}_p$ the current state of video class $p \in \mathcal{P}$ at TTI t. The controllable sub-state is defined as: $\mathbf{y}_p = [\mathbf{x}_p, \underline{\mathbf{x}}_p, \mathbf{q}_p]$, where $\underline{\mathbf{x}}_p = [\underline{\mathbf{x}}_{p,l_1}, \underline{\mathbf{x}}_{p,l_2}, ..., \underline{\mathbf{x}}_{p,l_{L_p}}]$, $\underline{\mathbf{x}}_{p,l} = [\underline{x}_{p,l,o_1}, \underline{x}_{p,l,o_2}, .., \underline{x}_{p,l,o_{O_p}}]$ and $\underline{x}_{p,l,o}$ is the difference between KPI value $x_{p,l,o}$ and its associated requirement $\bar{x}_{p,l,o}$ calculated for each QoS objective differently. One important objective is to ensure a Guaranteed Bit Rate (GBR) to each requested video service. Another important QoS objective is to keep the delay of video packets waiting in the queues under a certain threshold specific for each class. Last but not least, each delivered service must respect a given Packet Loss Rate (PLR) requirement for an appropriate video experience. If we consider $o_1 = GBR$, $o_2 = DELAY$ and $o_3 = PLR$, then we have:

$$\underline{x}_{p,l,o} = \begin{cases} \bar{x}_{p,l,o} - x_{p,l,o}, & o = o_1, \bar{x}_{p,l,o} > x_{p,l,o}, \\ x_{p,l,o} - \bar{x}_{p,l,o}, & o \in \{o_2, o_3\}, x_{p,l,o} > \bar{x}_{p,l,o}, \\ 0, & otherwise. \end{cases} \quad (2)$$

The vector $\mathbf{q}_p(t) = [\mathbf{q}_{p,l_1}, \mathbf{q}_{p,l_2}, ..., \mathbf{q}_{p,l_{L_p}}]$ represents the amount of data in each learner queue.

At the macro level with P number of video classes, the overall state at TTI t becomes $\mathbf{s}(t) = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_P] \in \mathcal{S}$, where $\mathcal{S}$ is the controller state space with the dimension of $\mathcal{S}_p$ multiplied by P. Each master and slave agent $p \in \mathcal{P}$ takes decisions based on its own state $\mathbf{s}_p \in \mathcal{S}_p$, by making the proposed framework a decentralized approach.

B. Action Space

We define by $\mathcal{A}_M$ the P dimensional action space of the master controller, where $\mathbf{c}(t) = [c_1, c_2, ..., c_P] \in \mathcal{A}_M$ is the current action at TTI t. Each element in the prioritization sequence is given by $c_k \in \mathcal{P}$, $1 \leq k \leq P$ and $c_k \neq c_{k'}$, $\forall k \neq k'$. Here, parameter $k \in \{1, 2, ..., P\}$ denotes the element index in the prioritization sequence $\mathbf{c}(t)$ decided at each TTI t, while $p \in \{1, 2, ..., P\}$ is the prioritization index given by 3GPP [31]. By $\mathcal{A}_S$ we define the action space of the slave controller with the action at TTI t defined as $\mathbf{r}(t) = [r_1, r_2, ..., r_P] \in \mathcal{A}_S$, where $r_p \in \mathcal{R}$ and $1 \leq p \leq P$.

We consider now $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times ... \times \mathcal{A}_P$ the overall action space, where $\mathcal{A}_k = \mathcal{P} \times \mathcal{R}$, $1 \leq k \leq P$. The action at TTI t becomes $\mathbf{a}(t) = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_P]$, and each $\mathbf{a}_k(t) = [c_k, \mathbf{r}(c_k)]$ gives the k-th instantaneous priority to video class $c_k \in \mathcal{P}$ to be scheduled and selects the scheduling rule $\mathbf{r}(c_k) \in \mathcal{R}$ given by slave agent c_k to perform the resource allocation. In Fig. 6, $c_1 = p \in \mathcal{P}$ is the first prioritized class while $\mathbf{r}(c_1) = r_p \in \mathcal{R}$ is the scheduling rule selected for $p \in \mathcal{P}$. By following the same reasoning, $c_P = 1 \in \mathcal{P}$ is the last prioritized class while $\mathbf{r}(c_P) = r_1 \in \mathcal{R}$ is the scheduling rule decided by slave agent 1. However, due to the network conditions of each class and the limited bandwidth, in some cases, only P^* number of classes are passed in the frequency domain, while mobile learners in other classes remain un-allocated at TTI t.

C. Transition Functions

Given the actual state $\mathbf{s}(t) \in \mathcal{S}$ and action $\mathbf{a}(t) \in \mathcal{A}$, the controllable state $\mathbf{y}' = \mathbf{y}(t+1) \in \mathcal{S}^C$ at TTI t+1 can be modeled as a function of $\mathbf{s}$ and $\mathbf{a}$. Specifically, if we consider $\mathbf{y}'_{\mathbf{a}} \in \mathcal{S}^C$ the controllable state at TTI $t+1$ as the result of applying action $\mathbf{a}(t)$ in state $\mathbf{s}(t)$, then the transition function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}^C$ can be written a follows:

$$\mathbf{y}'_{\mathbf{a}} = f(\mathbf{s}, \mathbf{a}), \quad (3)$$

where $\mathbf{y}'_{\mathbf{a}} = [\mathbf{y}'_{c_1}, \mathbf{y}'_{c_2}, ..., \mathbf{y}'_{c_P}]$ is vector $\mathbf{y}' = [\mathbf{y}'_1, \mathbf{y}'_2, ..., \mathbf{y}'_P]$ reordered according to the prioritization sequence decided at TTI t. The transition function can be decomposed as $f = f_{c_1} \circ f_{c_2} \circ ... \circ f_{c_P}$, where $\circ$ is a repetitive operator that allocates the remaining bandwidth once class $c_k \in \mathcal{P}$ has been scheduled, and $f_{c_k} : \mathcal{S}_{c_k} \times \mathcal{A}_k \to \mathcal{S}_{c_k}^C$. Then, any controllable element can be determined based on:

$$\mathbf{y}'_{c_k} = f_{c_k}[\mathbf{s}_{c_k}, \mathbf{a}_k]. \quad (4)$$

D. Reward Functions

The goal of the reward function is to measure at each TTI the multi-class and multi-objective QoS performance when applying the action $\mathbf{a}(t) \in \mathcal{A}$ in state $\mathbf{s}(t) \in \mathcal{S}$. Following the original definition from [41], this function becomes:

$$\rho(\mathbf{s}, \mathbf{a}) \overset{(\mathbf{def})}{=} \mathbb{E}\Big[\rho_{t+1} | \mathbf{s}(t) = \mathbf{s}, \mathbf{a}(t) = \mathbf{a}\Big], \quad (5)$$

where ρ_{t+1} is the reward at TTI t+1 and $\mathbb{E}[\cdot]$ is the expectation showing that $\mathbf{s}(t) \in \mathcal{S}$ is considered as random such that $\mathbb{P}[\mathbf{s}(t) = \mathbf{s}] > 0$ and $\mathbb{P}[\mathbf{a}(t) = \mathbf{a}] > 0$ holds for all possible actions $\mathbf{a} \in \mathcal{A}$. By employing the transition function from (3) and splitting the action in master and slave decisions, the reward function can be written as follows [23]:

$$\rho(\mathbf{s}, \mathbf{a}) \overset{(3)}{=} \rho(\mathbf{y}', \mathbf{a}) = \sum\nolimits_{k=1}^{P} \chi(c_k) \cdot \varrho_{c_k}[\mathbf{y}'_{c_k}, \mathbf{r}(c_k)], \quad (6)$$

where $\chi : \mathcal{P} \to \mathbb{R}$, $\chi(c_k) = (P+1-c_k)/\sum_{h=1}^{P} h$ is the weight function and $\varrho_{c_k} : \mathcal{S}_{c_k} \times \mathcal{R} \to [0, 1]$ is the reward of slave agent

$c_k \in \mathcal{P}$ that measures the impact of applying scheduling rule $\mathbf{r}(c_k) \in \mathcal{R}$ in state $\mathbf{s}_{c_k} \in \mathcal{S}_{c_k}$, while respecting the property of $\varrho_{c_k}[\mathbf{s}_{c_k}, \mathbf{r}(c_k)] \overset{(4)}{=} \varrho_{c_k}[\mathbf{y}'_{c_k}, \mathbf{r}(c_k)]$ with $\mathbf{s}_{c_k} = \mathbf{s}_p$ the state of class $p \in \mathcal{P}$ being scheduled at TTI t with the $k-$th priority.

Although the rewards ϱ in (6) are calculated in the same way, their meaning is different for the video classes in $\mathcal{P}^*$ and $\mathcal{P}^{\otimes}$. On one side, reward $\varrho_{c_{k^*}}[\mathbf{y}'_{c_{k^*}}, \mathbf{r}(c_{k^*})]$ measures the impact of using rule $\mathbf{r}(c_{k^*}) \in \mathcal{R}$ in class $\forall c_{k^*} \in \mathcal{P}^*$ and $k^* = 1, 2, ..., P^*$. On the other side, $\varrho_{c_{k^{\otimes}}}(\mathbf{y}'_{c_{k^{\otimes}}})$ evaluates if the objectives $\mathcal{O}_{c_{k^{\otimes}}}$ are met when the traffic class $c_{k^{\otimes}} \in \mathcal{P}^{\otimes}$ is not scheduled, and $k^{\otimes} = P^* + 1, ..., P$. By weighting the QoS revenues of each mobile learner, we have:

$$\varrho_p(\mathbf{y}'_p) = \sum\nolimits_{l=l_1}^{l_{L_p}} \frac{1}{L_p} \cdot \varrho_{p,l}(\mathbf{y}'_{p,l}), \tag{7}$$

where $\mathbf{y}'_{p,l} = [\mathbf{x}'_{p,l}, \underline{\mathbf{x}}'_{p,l}, q'_{p,l}]$ are the controllable state elements of learner $l \in \mathcal{L}_p$ at TTI $t+1$. The reward of each mobile learner is computed by considering the revenues of each QoS objective, such as:

$$\varrho_{p,l}(\mathbf{y}'_{p,l}) = \frac{1}{O_p} \sum\nolimits_{o=o_1}^{o_{O_p}} \varrho_{p,l,o}(\mathbf{y}'_{p,l,o}), \tag{8}$$

and $\mathbf{y}'_{p,l,o} = [x'_{p,l,o}, \underline{x}'_{p,l,o}, q'_{p,l}]$. For the particular case of $O_p = 3$, the sub-rewards $\varrho_{p,l,o}$ are calculated as follows:

$$\varrho_{p,l,o} = \begin{cases} 1 - \frac{\underline{x}_{p,l,o}}{\bar{x}_{p,l,o}}, & o = o_1, \underline{x}_{p,l,o} > 0, q_{p,l} \neq 0, \\ 1 - \frac{\underline{x}'_{p,l,o}}{x_{p,l,o}}, & o = \{o_2, o_3\}, \underline{x}_{p,l,o} > 0, q_{p,l} \neq 0, \\ 1, & otherwise. \end{cases} \tag{9}$$

When $\varrho_{p,l,o} = 1$, then objective $o \in \mathcal{O}$ of leaner $l \in \mathcal{L}_p$ requesting video content from class $p \in \mathcal{P}$ is met. The HiMARL framework aims to decide the multi-dimensional action $\mathbf{a}(t)$ in each state $\mathbf{s}(t)$ such that the total reward in (6) reaches its maximum level at each TTI.

The amount of inter-class fairness depends on the traffic load for each video class that may change over time. When the traffic load is high for all classes, the network gets saturated and the corresponding sub-rewards are $\varrho(\mathbf{y}'_p) < 1$ for all $p \in \mathcal{P}$. In this case, the inter-class fairness could be ensured by prioritizing classes while respecting the order given by $\chi(c_k)$. In other cases when the reward $\rho(\mathbf{s}, \mathbf{a})$ can still be maximized due to favourable network conditions and traffic load, the lower prioritized classes with $\varrho(\mathbf{y}'_{p'}) < 1$ could be prioritized at first in the detriment of other higher prioritized classes with $\varrho(\mathbf{y}'_p) = 1$, where $p < p' \in \mathcal{P}$. In this case, the inter-class fairness given by $\chi(c_k)$ is not respected anymore since the goal of the master controller is to maximize the overall QoS outcome at each TTI.

E. Policies and Value Functions

Based on the elements introduced in the previous sections, we define now a stochastic game for the slave controller as a tuple formed by $\langle \mathcal{S}_1, ..., \mathcal{S}_P, \mathcal{A}_1, ..., \mathcal{A}_P, f_1, f_2, ..., f_P, \rho \rangle$ with P number of fully cooperative intelligent agents since each one acts to maximize a common reward ρ [42]. Each slave agent $p \in \mathcal{P}$ is committed to learn over time its own policy function $\pi_p : \mathcal{S}_p \times \mathcal{R} \rightarrow [0, 1]$, that is, the probability of selecting scheduling rule $r_p \in \mathcal{R}$ in state $\mathbf{s}_p \in \mathcal{S}_p$ [41]. At each TTI, each agent $p \in \mathcal{P}$ decides to apply the scheduling rule with the highest policy value from vector $\pi_p(t) = [\pi_{p,1}, ..., \pi_{p,R}]$. By considering each class policy values π_p, we get the sequence:

$$\pi(t) = [\pi_1, \pi_2, ..., \pi_P].$$

The goal of the slave controller is to learn over time the optimal sequence of policies $\pi^*(t)$ to be followed such that the optimal action $\mathbf{r}^*(t) = [r_1^*, r_2^*, ..., r_P^*]$, $\forall r_p^* \in \mathcal{R}$ of all slave agents would have the highest QoS revenue at each TTI. However, training such MARL system is difficult to achieve in practice since the agents with lower priority are updated much less than other agents that represent the video classes with higher priorities from $\mathcal{P}$. To avoid this drawback, we propose to train each slave agent separately based on the tuple $\langle \mathcal{S}_p, \mathcal{R}, f_p, \varrho_p \rangle$. Each agent learns the best scheduling rule $r_p \in \mathcal{R}$ to be employed on each particular state $\mathbf{s}_p \in \mathcal{S}_p$ from the perspective of reward ϱ_p with the premise that the entire bandwidth is available for video class $p \in \mathcal{P}$. Precisely, we are interested in computing the optimal functions that map states $\mathbf{s}_p \in \mathcal{S}_p$ to actions $r_p \in \mathcal{R}$. To do so, value-functions-based RL are needed while considering the past experiences of each slave agent $p \in \mathcal{P}$. One of such functions is the action-value function $Q_p(\mathbf{s}_p, r_p)$ defined as the expected cumulative discounted future reward if agent is in state $\mathbf{s}_p \in \mathcal{S}_p$, executes action $r_p \in \mathcal{R}$, and follows the policy π_p afterwards [41]:

$$Q_p(\mathbf{s}_p, r_p) = \mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty} \gamma^t \varrho_{t+1} | \mathbf{s}_p(0) = \mathbf{s}_p, r_p(0) = r_p, \pi_p \Big], \tag{10}$$

where $0 \leq \gamma \leq 1$ is a discount factor that gives more importance to the immediate rewards than that to the later ones. In optimal case, each agent $p \in \mathcal{P}$ selects the optimal scheduling rule with probability $\pi_p^*(\mathbf{s}_p, r_p) = 1$ and,

$$r_p = argmax_{r' \in \mathcal{R}} Q_p^*(\mathbf{s}_p, r'). \tag{11}$$

A value function $V_p(\mathbf{s}_p)$ can provide the expected cumulative discounted future reward if the slave agent $p \in \mathcal{P}$ is in state $\mathbf{s}_p \in \mathcal{S}_p$ and underlies policy π_p afterwards [41]:

$$V_p(\mathbf{s}_p) = \mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty} \gamma^t \varrho_{t+1} | \mathbf{s}_p(0) = \mathbf{s}_p, \pi_p \Big]. \tag{12}$$

By extracting the reward at TTI $t = 0$ from (12), we get the following transition between two consecutive states $\mathbf{s}_p$ and $\mathbf{s}'_p$:

$$V_p(\mathbf{s}_p) = \varrho_p(\mathbf{s}_p, r_p) + \gamma \cdot V_p(\mathbf{s}'_p). \tag{13}$$

When training the master controller separately, the slave actions $\mathbf{r} \in \mathcal{A}_S$ are already known on each particular state and then, the overall action becomes $\mathbf{a}(t) \approx \mathbf{c}(t) \in \mathcal{A}_M$. The scope is to decide the traffic prioritization each TTI while considering the stochastic game with the tuple $\langle \mathcal{S}_1, ..., \mathcal{S}_P, \mathcal{A}_M, f, \rho \rangle$. Here, each master agent $p \in \mathcal{P}$ learns based on its own state space $\mathcal{S}_p$ to cooperate with other agents to maximize the total QoS revenue given by the reward in (6). Each master agent $p \in \mathcal{P}$ keeps its own policy $\Pi_p : \mathcal{S}_p \times \mathcal{A}_M \rightarrow [0, 1]$ that gives the probability of applying the prioritization sequence $\mathbf{c} \in \mathcal{A}_M$ in state $\mathbf{s}_p \in \mathcal{S}_p$. Then, the joint policy of master controller can be defined as the following sequence:

$$\Pi(\mathbf{c}) = [\Pi_1, \Pi_2, ..., \Pi_P].$$

Together with policy Π_p, each agent keeps an action-value function $W_p(\mathbf{s}_p, \mathbf{c})$ that represents the expected cumulative discounted future reward if agent $p \in \mathcal{P}$ is in state $\mathbf{s}_p$, executes

the joint action $\mathbf{c} \in \mathcal{A}_M$ by getting the k-th priority to be scheduled, and the joint policy Π is followed afterwards:

$$W_p(\mathbf{s}_p, \mathbf{c}) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \rho_{t+1} | \mathbf{s}_p(0) = \mathbf{s}_p, \mathbf{c}(0) = \mathbf{c}, \Pi\Big]. \quad (14)$$

Under optimal conditions, an action $\mathbf{c} \in \mathcal{A}_M$ is selected with a sequence of probabilities $\Pi(\mathbf{c}) = [1, 1, ..., 1]$ and:

$$\mathbf{c} = \mathbf{solve}_{p \in \mathcal{P}}[W_p^*(\mathbf{s}_p, \cdot)]_{p=1,2,...,P}, \quad (15)$$

where $\mathbf{solve}$ performs the descent ordering of all action-values and returns the arguments of the obtained sequence.

In our approach, we make also use of value-function $V(\mathbf{s})$ of the entire state space that starts with the initial state $\mathbf{s}(0) = \mathbf{s} \in \mathcal{S}$ and underlies the joint policy Π afterwards:

$$V(\mathbf{s}) = \mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty} \gamma^t \rho_{t+1} | \mathbf{s}(0) = \mathbf{s}, \Pi\Big]. \quad (16)$$

The role of $V(\mathbf{s})$ is to enhance the training process of the master controller and to coordinate the master agents to update their action-value functions when the applied actions have positive impact in the system performance. Moreover, the transition between two consecutive states can be used as well:

$$V(\mathbf{s}) = \rho(\mathbf{s}, \mathbf{c}) + \gamma \cdot V(\mathbf{s}'). \quad (17)$$

The HiMARL framework presented above is more conceptual as long as the state space $\mathcal{S}_p$ of each agent is continuous with a variable dimension given the number of active learners of each video class $p \in \mathcal{P}$. Hence, it is impossible to exhaustively enumerate all possible combinations of state-action pairs and to store the value functions. To overcome some of these problems, in a first instance, each agent state space must be transformed to a compact representation with a constant dimension. In the second instance, the function approximators must be employed and trained to approximate the optimal value functions of master and slave controllers.

F. State Space Compression

The dimension of the current state $\mathbf{s} \in \mathcal{S}$ depends on the number of active mobile learners that may change over time for each traffic class $p \in \mathcal{P}$. This aspect leads to a variable dimension of the entire state space $\mathcal{S}$ that contains observations collected from the interaction between the intelligent controller and network. To reduce the framework complexity, the variable dimension must be reduced to some constant representation regardless of the changeable networking conditions. Since the proposed system takes decisions separately based on the observations collected from each traffic class, then the compression procedure involves the following transformation:

$$\bar{\mathcal{S}}_p = \mathcal{T}(\mathcal{S}_p), \quad (18)$$

where $\bar{\mathcal{S}}_p$ is the compressed state space of class $p \in \mathcal{P}$ with a fixed dimension. Since the original state space is divided into controllable and uncontrollable spaces, then the compression framework should follow the same trend.

We denote by $\bar{\mathcal{S}}_p^C = \mathcal{T}_c(\mathcal{S}_p^C)$ the compressed controllable state space as the result of the compression operator $\mathcal{T}_c$ over the original version $\mathcal{S}_p^C$. The instantaneous controllable state becomes $\bar{\mathbf{y}} = \tau_c(\mathbf{y})$, where $\bar{\mathbf{y}} \in \bar{\mathcal{S}}_p^C$ and τ_c can be a set of statistical functions employed to extract the relevant data from a vector of L_p elements. We adopt the same statistical model proposed in [23], where the controllable elements $\{\mathbf{x}_p, \underline{\mathbf{x}}_p, \mathbf{q}_p\}$ can be modeled as normally distributed variables and then, τ_c employs the mean and standard deviation functions based on the maximum likelihood estimators. As result, the dimension of $\bar{\mathbf{y}}_p$ is reduced from $L_p \cdot (2O_p + 1)$ to $2 \cdot (2O_p + 1)$, that depends only on O_p number of objectives that must be met to keep high service quality for the delivered educational content.

A different compression approach should be employed for the uncontrollable space $\bar{\mathcal{S}}_p^U = \mathcal{T}_u(\mathcal{S}_p^u)$, where $\mathcal{T}_u$ differs from $\mathcal{T}_c$ due to the CQI information which is considered as a bandwidth dependent vector that is periodically reported by each mobile learner to inform the BS on its channel quality. Let us assume that $\mathbf{v}_p \in \mathcal{S}_p^U$ refers strictly to the CQI observations for class $p \in \mathcal{P}$. This uncontrollable vector takes the form of $\mathbf{v}_p = [\mathbf{v}_{p,1}, ..., \mathbf{v}_{p,L_p}]$ where $\mathbf{v}_{p,l} = [cqi_{p,l,1}, ..., cqi_{p,l,B}]$ and $cqi_{p,l,b} \in \{1, ..., 15\}$ for OFDMA radio access. To compress the large dimension of $\mathbf{v}_p$, we employ the compression function τ_u, where $\bar{\mathbf{v}}_p = \tau_u(\mathbf{v}_p)$. Here, τ_u aims to classify and predict each CQI report $\mathbf{v}_{p,l}$ in patterns or CQI clusters based on unsupervised and supervised learning techniques [40]. If we denote by Z the number of CQI classes to best represent the space $\mathcal{S}_p^U$, then according to τ_u, the dimension of $\mathbf{v}_p \in \mathcal{S}_p^U$ is reduced from $L_p \cdot B$ to Z, where $L_p > 1$ and $Z < B$.

To conclude, a compression mechanism is possible in order transform the state representation from $\mathbf{s}_p \in \mathcal{S}_p$ to $\bar{\mathbf{s}}_p \in \bar{\mathcal{S}}_p$ and its dimension from $L_p \cdot (2O_p + 1 + B)$ to $2 \cdot (2O_p + 1) + Z$.

G. Function Approximation with Reinforcement Learning

Once applied the compression mechanism, the entire state $\bar{\mathbf{s}} \in \bar{\mathcal{S}}$ is still multi-variable but, with a fixed dimension. Therefore, the classical tabular representation of value functions is not possible and function approximations must be adopted instead for both master and slave controllers. In this paper, we adopt the use of feed-forward neural networks to represent such approximations. However, more complex architectures of neural networks can be employed by considering the time constraint given by the TTI duration. Without going through very precise details about the internal structure of a feed-forward neural network, we resume to consider such a structure as a parameterizable function $F(\cdot; \theta)$, where θ represents the set of interconnecting weights between the internal nodes that must be optimized by the proposed RL and HiMARL strategies.

Considering now the training process of the slave controller, neural networks are used to approximate the look-up table of Q-values for each agent. We focus only on a particular slave agent, let us say, $p \in \mathcal{P}$, while the rest of them follows exactly the same reasoning. Slave agent $p \in \mathcal{P}$ is characterized by R number of neural networks, where each such entity approximates the action-value function for a given scheduling rule $r_p \in \mathcal{R}$. Let $Q_p(\mathbf{s}_p, r_p) \approx Q_p(\bar{\mathbf{s}}_p, r_p; \theta_{p,r})$ be such approximations, where $\theta_{p,r}$ denotes the set of weights corresponding to the neural network that represents the scheduling rule $r_p \in \mathcal{R}$. To improve the training quality of our framework, the additional approximation of value function is considered, such as: $V_p(\mathbf{s}_p) \approx V_p(\bar{\mathbf{s}}_p; \theta_p)$. The entire set of weights $\{\theta_{p,1}, \theta_{p,2}, ..., \theta_{p,R}, \theta_p\} \in [-1, 1]$ are randomized at the beginning and they are expected to converge to optimal

values as the training process evolves. During training, the exploration of the state space is achieved by picking at each TTI an action with the following probability [43]:

$$\pi_p = \begin{cases} 1-\epsilon & r_p = argmax_r Q(\bar{\mathbf{s}}_p, r_p; \theta_{p,r}), \\ \epsilon & r_p = argmax_r[\mathbf{rand}_r]_{r=1,2,...,R}, \end{cases} \quad (19)$$

where $\mathbf{rand}_r$ gives a set of random action-values to enhance the problem exploration. Parameter $\epsilon \in [0,1]$ can be varied from high to low during training to permit more explorations at the beginning and more exploitation of Q-functions as the training process approaches it its end. Once the action $r_p \in \mathcal{R}$ is decided in state $\bar{\mathbf{s}}_p \in \bar{\mathcal{S}}_p$, the system gets the reward value $\varrho_p(\bar{\mathbf{s}}_p, r_p)$ and a new state $\bar{\mathbf{s}}'_p \in \bar{\mathcal{S}}_p$ is perceived. Therefore, at each TTI $t+1$, we use the experience set $e(t+1) = \{\bar{\mathbf{s}}_p, r_p, \varrho_p, \bar{\mathbf{s}}'_p\}$ to update the neural network weights of slave each agent $p \in \mathcal{P}$.

The objective in the training process is to minimize over time the following cost function [44]:

$$\mathcal{C}_p(\theta_p) = \mathbb{E}_{e(t)}\Big[\frac{1}{2}(\eta\delta_p)^2\Big], \quad (20)$$

where $\eta \in [0,1]$ is the learning rate and $\delta_p = F_p^T(\cdot;\theta_p) - F_p(\cdot;\theta_p)$ is the Time Difference (TD) error calculated as the difference between the target value and the actual estimate of the neural network. The TD error is back-propagated through the neural networks to update the weights [23].

Conceptually, the weights of each layer are updated through the online training process while employing the Stochastic Gradient Descent (SGD) algorithm [43]:

$$\theta_p \leftarrow \theta_p + \eta \frac{\partial F_p}{\partial \theta_p}(\cdot;\theta_p) \cdot \delta_p, \quad (21)$$

where F_p takes the form of value and action-value functions. The target value function is determined according to (13) and becomes: $V_p^T(\bar{\mathbf{s}}_p;\theta_p) = \varrho_p + \gamma \cdot V_p(\bar{\mathbf{s}}'_p;\theta_p)$. However, when the TD error of value function is $\delta_p \geq 0$, then the action $r_p \in \mathcal{R}$ applied in state $\bar{\mathbf{s}}_p$ is a good option and such actions should be used in future. To do so, we set the target function of the action-value with the corresponding learning rates as follows:

$$Q_p^T(\bar{\mathbf{s}}_p, r_p; \theta_{p,r}) = \begin{cases} 1\ with\ \eta = \alpha & if\ \delta_p \geq 0, \\ -1\ with\ \eta = \beta & if\ \delta_p < 0, \end{cases} \quad (22)$$

where $\{\alpha, \beta\} \in [0,1]$ are two possible values for the learning rate η, constrained always by $\alpha > \beta$. When $\delta_p \geq 0$, agent $p \in \mathcal{P}$ learns more from such positive experiences and consequently, $\eta = \alpha$. When $\delta_p < 0$, the proposed algorithm learns less from the negative experiences and $\eta = \beta$. It is worth to notice that the same RL algorithm is followed by each slave agent $p \in \mathcal{P}$ to learn its own policy π_p^* based on the video traffic characteristics and network conditions. The principles of RL training for the slave agent $p \in \mathcal{P}$ are detailed by Algorithm 1.

The master controller makes use of P number of action-value functions that are parameterized with feed-forward neural networks, such as: $W_p(\bar{\mathbf{s}}_p, \mathbf{c}) \approx W_p(\bar{\mathbf{s}}_p, \mathbf{c}; \Theta_p)$, $p = 1, 2, .., P$. Compared to slave controller where the action-value functions are used to approximate the look-up tables, here these functions are trained to provide the prioritization decision at each TTI. To increase the efficiency of our framework,

Algorithm 1: RL Training of Slave Agent $p \in \mathcal{P}$

for each TTI t+1
  calculate the reward $\varrho_p(\mathbf{s}_p, r_p)$ based on (7)-(9)
  recall the experience $e_p(t+1) = (\mathbf{s}_p, r_p, \varrho_p, \mathbf{s}'_p)$
  compress $\{\mathbf{s}_p, \mathbf{s}'_p\} \in \mathcal{S}_p$ based on (18)
  calculate the value function error $\delta_p(\theta_p)$
  back-propagate $\delta_p(\theta_p)$ and update weights based on (21)
  // criticize previous action $r_p \in \mathcal{R}$
  if $\delta_p(\theta_p) \geq 0$, then $\eta = \alpha$, else $\eta = \beta$
  determine the target function based on (22)
  calculate error $\delta_{p,r}(\theta_{p,r})$
  back-propagate $\delta_{p,r}(\theta_{p,r})$ and update $\theta_{p,r}$ based on (21)
  // act based on the learned policy
  apply the scheduling rule $r'_p \in \mathcal{R}$ with policy (19)
  perform resource allocation according to selected $r'_p \in \mathcal{R}$
end for

we employ the value function given the entire state $\bar{\mathbf{s}} \in \bar{\mathcal{S}}$ and approximated by $V(\bar{\mathbf{s}}) \approx V(\bar{\mathbf{s}};\Theta)$. Similar to the slave controller, we assume the number of layers and nodes are fixed and the activation functions of each layer known. To this extent, we denote by $\{\Theta_1, \Theta_2, ..., \Theta_P, \Theta\}$ as the set of weights that must be tuned during the training process of the master controller. In the training stage, each agent takes a joint action $\mathbf{c} \in \mathcal{A}_M$ according the following policy:

$$\Pi_p(\mathbf{c} \mid \bar{\mathbf{s}}_p) = \begin{cases} 1-\epsilon & \mathbf{c} = \mathbf{solve}[W_p(\cdot;\Theta_p)]_{p=1,..,P}, \\ \epsilon & \mathbf{c} = \mathbf{solve}[\mathbf{rand}_p]_{p=1,..,P}, \end{cases} \quad (23)$$

where $\mathbf{rand}_p \in [0,1]$ is a sequence of random numbers and $\mathbf{solve}$ performs the descent ordering of a given sequence. Please note that ϵ may vary during the training process but the same value is used by all agents at each TTI.

At TTI t, we apply the joint action $\mathbf{c} \in \mathcal{A}_M$ in state $\bar{\mathbf{s}} \in \bar{\mathcal{S}}$, the scheduler grants this decision with reward $\rho(\bar{\mathbf{s}}, \mathbf{c})$ and new observations $\bar{\mathbf{s}}' \in \bar{\mathcal{S}}$ are perceived at TTI $t+1$. Let us define by $E(t+1) = \{\bar{\mathbf{s}}, \mathbf{c}, \rho, \bar{\mathbf{s}}', P^*(t)\}$ the experience of the master controller at TTI $t+1$, where $P^*(t)$ is the number of video classes being scheduled at TTI t. The master agent $p \in \mathcal{P}$ experiences the following set $E_p(t+1) = \{\bar{\mathbf{s}}_p, \mathbf{c}, \rho, \bar{\mathbf{s}}'_p\}$. Experiences $\{E_1, E_2, ..., E_P, E\}$ are used at each TTI with the scope of minimizing the following cost function of each agent:

$$\mathcal{C}(\Theta) = \mathbb{E}_{E(t)}\{\frac{1}{2}[\eta \cdot \Delta(\Theta)]^2\}, \quad (24)$$

where $\Delta(\Theta) = F^T(\cdot;\Theta) - F(\cdot;\Theta)$ is the TD error with function $F(\cdot;\Theta)$ representing all $W_p(\cdot;\Theta_p)$ and $V(\cdot;\Theta)$. By back-propagating the TD error from layer-to-layer, the weights are updated with the same SGD principle as follows [43]:

$$\Theta \leftarrow \Theta + \eta \frac{\partial F}{\partial \Theta}(\cdot;\Theta) \cdot \Delta(\Theta), \quad (25)$$

where (25) is employed for each master agent $p \in \mathcal{P}$ and value function. The target value function is determined according to (17) and then, $V^T(\bar{\mathbf{s}};\Theta) = \rho + \gamma \cdot V(\bar{\mathbf{s}}';\Theta)$. The corresponding TD error becomes in this case $\Delta(\Theta) = V^T(\bar{\mathbf{s}};\Theta) - V(\bar{\mathbf{s}};\Theta)$. Logically, when $\Delta(\Theta) \geq 0$, then the previous experience has a positive impact on the overall structure of master controller. However, even if the action $\mathbf{c} \in \mathcal{A}_M$ has a positive impact, some traffic classes with $\varrho_p = 1$ (all QoS objectives are met) can get higher priority than that of other classes with $\varrho_p < 1$,

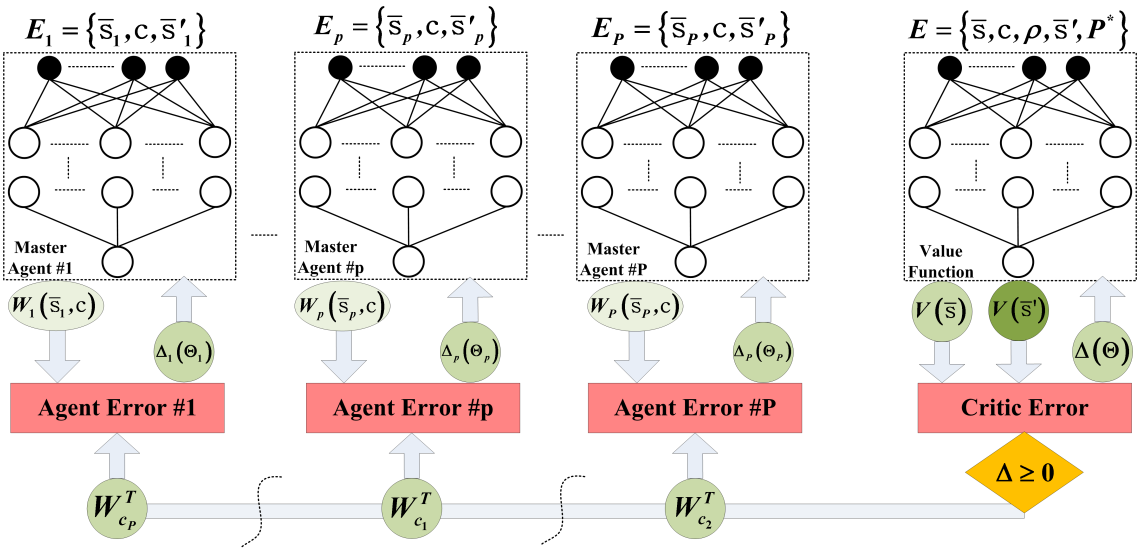


Fig. 7: Training process of the master controller

leading to the over-provisioning effect. To avoid this issue and efficiently allocate the available bandwidth, the master controller must be trained to avoid prioritizing video classes with fulfilled QoS requirements over other classes where the set of QoS requirements is partially met. To this respect, we employ the following function $h : \mathcal{P}^* \times [0,1]^P \rightarrow \{0,1\}$, such that: when $h(c_{k^*}, \varrho_1, ..., \varrho_P) = 1$, then traffic class $c_{k^*} = p^* \in \mathcal{P}^*$ respects all QoS requirements and gets higher priority than that of other classes that need more resources to meet their QoS requirements; when $h(c_{k^*}, \varrho_1, ..., \varrho_P) = 0$, then scheduling the traffic class $c_{k^*} = p^* \in \mathcal{P}^*$ with the $k^* - th$ priority is a fair option. Consequently, we employ the target action-value function as follows:

$$W^T_{c_{k^*}} = \begin{cases} \frac{P}{(P+1-k^*)} \; with \; \eta = \alpha & if \; \Delta \geq 0 \; and \; h(\cdot) = 0, \\ -0.5 \; with \; \eta = \alpha & if \; \Delta \geq 0 \; and \; h(\cdot) = 1, \\ -1 \; with \; \eta = \beta & if \; \Delta < 0, \end{cases} \tag{26}$$

where $W^T_{c_{k^*}}(\bar{\mathbf{s}}_{c_{k^*}}, \mathbf{c}; \Theta_{c_{k^*}})$ is the target action-value function of class $c_{k^*} \in \mathcal{P}^*$ being scheduled with the k^*-th priority at TTI t. If some traffic classes remain un-scheduled and $P^* < P$, then the afferent agents are not updated.

Even when the previous applied prioritization sequence has a positive impact in terms of heterogeneous QoS provisioning ($\Delta \geq 0$), the prioritization of class $c_{k^*} = p^* \in \mathcal{P}^*$ over other classes must be penalized when $h(c_{k^*}, \varrho_1, ..., \varrho_P) = 1$. In this way, the neural networks of the master controller are trained to avoid the over-provisioning effect between video classes with low and/or high bit-rates by considering the target values determined based on (26) at each TTI.

Figure 7 presents the MARL training principle of the master controller. At each TTI, experience $E(t+1)$ is used by the value function approximator (critic) to evaluate the quality of the joint action previously applied. Consequently, the values of two consecutive states $\{V(\bar{\mathbf{s}}), V(\bar{\mathbf{s}}')\}$ are determined and the corresponding TD error $\Delta(\Theta)$ is computed and back-propagated. Based on (26), a set of target action-value functions $\{W^T_{c_1}, W^T_{c_2}, ..., W^T_{c_P}\}$ is determined, meaning that

Algorithm 2: MARL Training of Master Controller

for each TTI t+1
  calculate $\varrho_p(\mathbf{s}_p, r_p)$ for each slave agent p based on (7)-(9)
  calculate $\rho(\mathbf{s}, \mathbf{c})$ according to (5) and (6)
  recall experiences $\{E_1, E_2, ..., E_P, E\}$
  compress states $\{\mathbf{s}_p, \mathbf{s}'_p\}_{p=1,...,P}$ and implicitly $\{\mathbf{s}, \mathbf{s}'\}$ - (18)
  calculate the value function error $\Delta(\Theta)$
  back-propagate $\Delta(\Theta)$ and update weights based on (25)
  // **criticize** previous action $\mathbf{c} \in \mathcal{A}_M$
  if $\Delta(\Theta) \geq 0$, then $\eta = \alpha$, else $\eta = \beta$
  for $k^* = 1, 2, ..., P^*$
    determine target function $W^T_{c_{k^*}}$ based on (26)
    calculate error $\Delta_{c_{k^*}}(\Theta_{c_{k^*}})$
    back-propagate $\Delta_{c_{k^*}}(\Theta_{c_{k^*}})$, update $\Theta_{c_{k^*}}$ based on (25)
  end for
  // **act** based on the joint policy
  determine new action $\mathbf{c}' \in \mathcal{A}_M$ based on policy (23)
  // prioritization and scheduling with $\mathbf{c}' \in \mathcal{A}_M$ and $\mathbf{r}' \in \mathcal{A}_S$
  while $\mathcal{B} \neq \varnothing$
    pick video class $c'_k = p$, $\forall p \in \mathcal{P}$
    exploit slave agent $c'_k = p$ and get $r'_p \in \mathcal{R}$
    perform scheduling and resource allocation based on (1)
    $k = k + 1$
    add $c'_k = p$ in the set of scheduled video classes $\mathcal{P}^*$
    $P^* = P^* + 1$
  end while
end for

$P = P^*$ in this particular case. Each master agent $p = c_k \in \mathcal{P}$ reinforces its TD error $\Delta_p(\Theta_p)$ calculated as the difference between the target value $W^T_{c_k}$ and the actual estimate W_{c_k}. To better highlight the concept of training the master controller, Algorithm 2 presents the most important steps.

VI. Results and Discussion

The proposed HiMARL framework is implemented in the 5G-Scheduler C/C++ Simulator [23] that inherits the LTE-Sim simulator [45] in terms of: protocol stack functionalities, OFDMA radio access technology, and scenario settings. The 5G-Scheduler simulator [23] brings some new features, such as: decoupled TDP and FDP schedulers, state space compres-

sion mechanisms as discussed in Section V.F, neural networks as function approximators and different RL algorithms. The goal of this section is to validate through objective numerical results the proposed HiMARL framework in order to make it a suitable option when scheduling mobile learners with different needs in terms of the educational content. To this extent, we organize this discussion as follows: *a)* first we present the parameter settings and the considered scenario; *b)* then, through various numerical results, the advantage of implementing the slave controller is presented; *c)* the performance of the HiMARL framework is analysed in the third instance based on the comparison with state-of-the-art schedulers; *d)* and finally, the remarks on the obtained subjective and objective assessments are analyzed in terms of mobile learner satisfaction.

A. Network and Controller Settings

The number of video classes to be analysed is fixed to $P = 4$. To determine the composition of each video class, the subjective results from Section III are combined with the findings from [26]. Moreover, the QoS requirements for the considered video content of each class are determined according to the 4G technical specifications [31].

According to figures 2, 3 and 4, a significant statistical difference between high and low quality can be observed in the case of slideshow content, while in the case of screencast and animation contents, the learning achievement level is near similar regardless of the video quality levels. We consider two types of devices with the following resolutions [26]: `class_a` 240p, denoting the class of mobile learners with reduced financial possibilities; `class_b` 480p, that represents the class of mobile learners with relatively better financial situations. The aim is to prioritize the first class of learners without penalizing too much the second class. To this extent, we consider that learners from the first class request the slideshow content with high and low quality, while learners from the second class access the animation and screencast contents at low quality. The animation video can be modeled as Variable Bit Rate (VBR) traffic and the screencast content as Constant Bit Rate (CBR). By correlating the information from [26] and [31], the following video classes with the corresponding QoS requirements are considered in terms of $o_1 = GBR$, $o_2 = DELAY$ and $o_3 = PLR$, respectively:

- $p = 1$: `video_1` (slideshow, high quality), $\bar{x}_{1,l,o_1} = 242kpbs$, $\bar{x}_{1,l,o_2} = 150ms$, and $\bar{x}_{1,l,o_3} = 10^{-3}$, $\forall l \in \mathcal{L}_1$;
- $p = 2$: `video_2` (slideshow, low quality), $\bar{x}_{2,l,o_1} = 138kpbs$, $\bar{x}_{2,l,o_2} = 300ms$, and $\bar{x}_{2,l,o_3} = 10^{-6}$, $\forall l \in \mathcal{L}_2$;
- $p = 3$: `video_3` (animation, low quality), $\bar{x}_{3,l,o_1} = 512 - 1024kpbs$, $\bar{x}_{3,l,o_2} = 300ms$, and $\bar{x}_{3,l,o_3} = 10^{-6}$, $\forall l \in \mathcal{L}_3$;
- $p = 4$: `video_4` (screencast, low quality), $\bar{x}_{4,l,o_1} = 640kpbs$, $\bar{x}_{4,l,o_2} = 300ms$, and $\bar{x}_{4,l,o_3} = 10^{-6}$, $\forall l \in \mathcal{L}_4$.

By employing the proposed HiMARL meta-scheduler, the aim is to create a dynamic prioritization and scheduling policy to maximize the QoS outcome of lower prioritized classes (i.e. `video_3` and `video_4`) without affecting the learning achievement levels of individuals with higher prioritized services, such as `video_1` and `video_2`.

TABLE III: **Parameter Settings**

Parameter	Value/Description
System Bandwidth/Cell Radius	20 MHz/1000m
Speed/Mobility (Learning)	30 Kmph/Random Direction
Speed/Mobility (Exploitation)	Static/Uniform Distribution
Channel Model	Jakes Model
Path Loss/Penetration Loss	Macro Cell Model/10dB
Interfered Cells/Shadowing STD	6/8dB
Carrier Frequency/DL Power	2GHz/43dBm(equal on each RB)
Frame Structure	Frequency Division Duplexing
CQI Reporting Mode	Full-band, periodic at each TTI
PUCCH Model	Errorless
PDSCH Model	Wideband Error Model based on Effective SINR [45]
Max. No. of Schedulable Users	32 each TTI
RLC ARQ	Acknowledged Mode (5 retransmissions)
AMC Levels	QPSK, 16-QAM, 64-QAM
Target BLER	10%
Traffic Type	Heterogeneous (16.5% `video_1`, 16.5% `video_2`, 33% `video_3`, 33% `video_4`)
QoS Requirements $(GBR, Delay, PLR, p)$	$(242kbps, 150ms, 10^{-3}, 1)$, `video_1` $(138kbps, 300ms, 10^{-6}, 2)$ `video_2` $(512 - 1024kbps, 300ms, 10^{-6}, 3)$ `video_3` $(640kbps, 300ms, 10^{-6}, 4)$ `video_4`
Max. No. of Users	10 (`video_1`), 10 (`video_2`) 20 (`video_3`), 20 (`video_4`)
Frequency Schedulers	SA_1, SA_2, SA_3, SA_4, BF [37], EXP [38], OPLF [39]
Time-Frequency Schedulers	HiMARL, RADS [33], FLS [34]
Learning/Exploitation Duration	10000s/10 $\times$ 51s
RL Discount Factor (γ)	0.99
Neural Networks Configurations	$(2\ layers, 80\ nodes)-$SC, $(2\ layers, 200\ nodes)-$MC.

On the network side, we consider the OFDMA downlink transmissions with 20MHz system bandwidth with a number of RBs $B = 100$. A macro urban cell model is considered with the radius of 1km. The channel model is fast fading by employing the Jakes model [45]. The inter-cell interference model with other neighboring cells is considered, while the intra-cell interference between other mobile and electronic equipment is considered negligible in this case. It is worth to mention that the performance of proposed HiMARL meta-scheduler is compared with state-of-the-art schedulers by using exactly the same networking conditions. More details on the scenario settings can be found in Table III.

The network scheduler works with the Modulation and Coding Scheme (MCS) that associates different levels of modulation schemes as shown in Table III necessary to transmit the quota of data scheduled to each mobile learner. The Radio Link Control (RLC) protocol works in the acknowledged mode and considers a maximum number of 5 re-transmissions for each erroneous packet for all video classes. Packets failing to get successfully transmitted within this period are declared lost. When computing the PLR rates and average user throughput, a moving time window of 1000 TTIs is used to collect the lost packets and instantaneous user throughput, respectively. Then, these KPI indicators are matched to the corresponding requirements for each class (Table III). In terms of the scheduling process, the following strategies are employed:

a) FDP Scheduling: the slave controller is employed to select for each of the considered video class the scheduling rule to be followed in the frequency domain to maximize the multi-objective QoS provisioning. To this extent, we consider

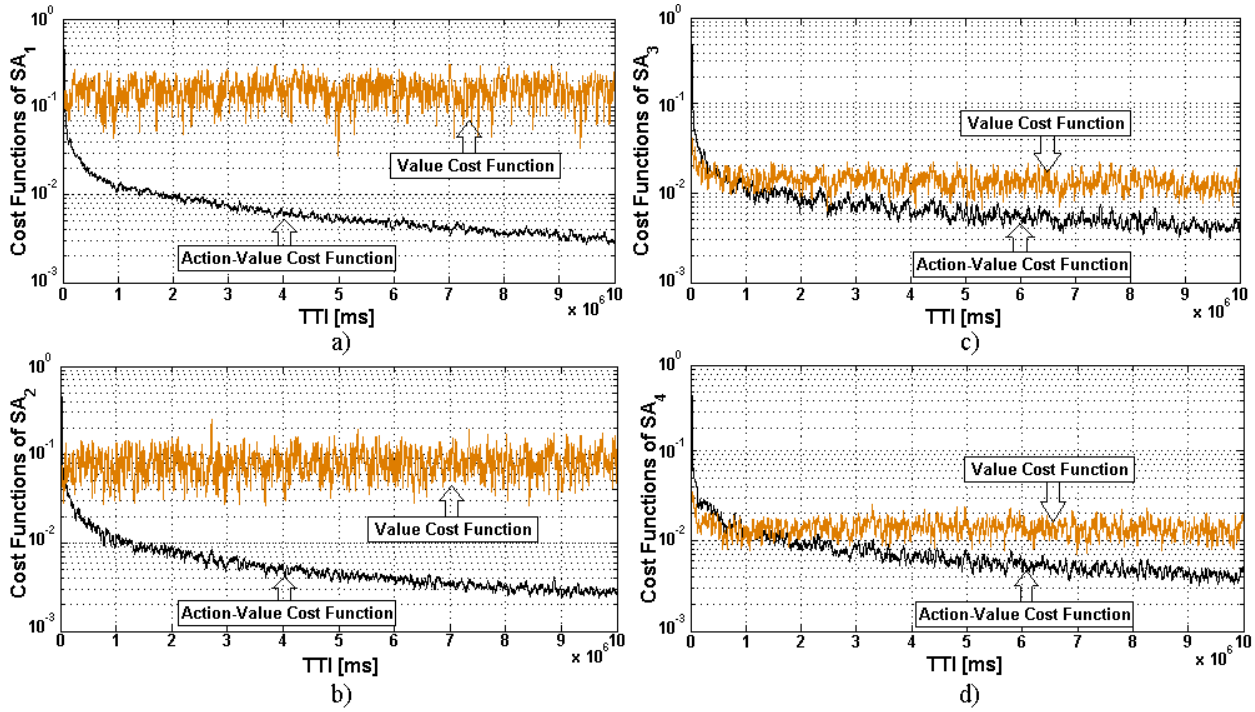


Fig. 8: Training stage of the slave controller a) cost function of slave agent 1 (`video_1`); b) cost function of slave agent 2 (`video_2`); c) cost function of slave agent 3 (`video_3`); d) cost function of slave agent 4 (`video_4`).

a number of $R = 3$ scheduling rules, being organized as follows: $r = 1$ corresponds to Barrier Function (BF) [37] oriented to meet the GBR objective, such that, the online average throughput indicator respects its requirement; $r = 2$ is the EXPonential (EXP) scheduling rule [38] employed to meet the delay requirements; $r = 3$ implies the Opportunistic Packet Loss Fair (OPLF) [39] scheduler oriented on minimizing the PLR rates. During training, each slave agent randomizes the number of active users in the following ranges: $L_1 \in [10, 50]$, $L_2 \in [10, 165]$, $L_3 \in [10, 35]$ and $L_4 \in [10, 35]$. Below the lowest limit in terms of the number of mobile learners $L_1 = L_2 = L_3 = L_4 = 10$, the QoS provisioning is mostly ensured and there is no advantage of HiMARL versus the alternative approaches. Above the maximum limits (i.e. $L_1 = 50$, $L_2 = 165$, $L_3 = 35$, $L_4 = 35$), the network gets saturated for the considered bandwidth, and the employment of any scheduler would not bring significant improvement when evaluating the QoS provisioning. During exploitation, the performance is evaluated for each number of mobile learners in the aforementioned intervals.

b) TDP Scheduling: considers the training and the exploitation of master controller; during training, the slave controller is exploited while, during the exploitation, the entire hierarchical MARL framework is employed for real-time scheduling. The aim of the master controller is to dynamically switch the prioritization sequence every TTI such that the QoS provisioning of `video_1`, `video_2`, `video_3` and `video_4` is maximized every time. In both training and exploitation stages, we consider the following ratio between the learner classes: `video_1` (16.5%), `video_2` (16.5%), `video_3` (33%), and `video_4` (33%). In training and exploitation stages, the aggregate number of mobile learners are varied in the interval of $l \in [6, 60]$, and the maximum number of learners of each video class are: $L_1 = 10$, $L_2 = 10$, $L_3 = 20$ and $L_4 = 20$. In exploitation, the performance of HiMARL framework is compared with very promising state-of-the-art strategies, such as: Required Activity Detection Scheduler (RADS) [33] and Frame Level Scheduler (FLS) [34].

The proposed HiMARL framework is trained in separate stages, but some parameters are similar for both slave and master controllers. The discount factor is set to $\gamma = 0.99$ to provide higher importance to the value of the next-states when computing the target functions as shown in (13) and (17). We aim to set the learning rate $\beta = 0$ when the experiences are not favorable ($\delta_p(\theta_p) < 0$, $\Delta_p(\Theta_p) < 0$). Learning rate α is decreased with a predefined step during the training process for each slave and master agent, each time when $\delta_p(\theta_p) \geq 0$ and $\Delta_p(\Theta_p) \geq 0$, respectively. Separately, the learning rates for slave and master value functions are decreased at each iteration. Parameter ϵ from (19) and (23) decreases exponentially from 1 to 0, such that, more exploration steps would be involved at the beginning of the training stage, while more exploitation steps would be used at the end of this process.

In terms of the evaluation metrics, the numerical results are organized as follows: *a) metrics for training:* we monitor over time the mean cost functions for value and action-value neural networks for the master controller and each slave agent; *b) metrics for exploitation:* the mean and STandard Deviation (STD) functions are implemented for each evaluation metric, such as:

$$\mu_p(\mathbf{m}_p) = 1/G \cdot \sum_{g=1}^{G} \mathbf{m}_{p,g}, \tag{27.a}$$

$$\sigma(\mathbf{m}_p) = \sqrt{1/G \cdot \sum_{g=1}^{G} (\mathbf{m}_{p,g} - \mu_p)^2}, \tag{27.b}$$

where $G = 10$ is the number of simulations in the exploitation stage and $\mathbf{m}_p$ is the metric that evaluates the performance of

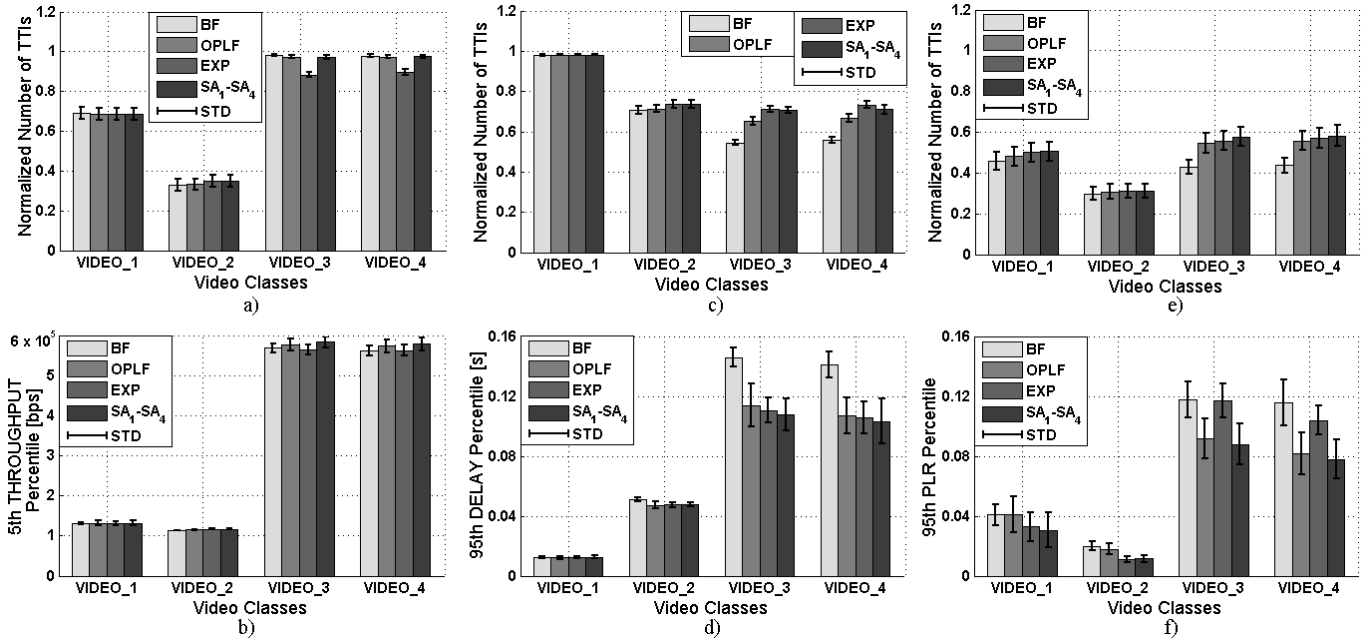


Fig. 9: Performance of the slave agents based on $\mu_p(\mathbf{m}_p)$ and $\sigma_p(\mathbf{m}_p)$, where metrics $\mathbf{m}_p$ are: a) normalized number of TTIs when all learners meet the GBR requirement; b) 5^{th} throughput percentile; c) normalized number of TTI when all learners respect the delay requirements; d) 95^{th} delay percentile; e) normalized number of TTIs when all learners meet the PLR requirements; f) 95^{th} PLR percentile.

the involved scheduling candidates for each video class $p \in \mathcal{P}$.

B. Training and Testing the Agents of the Slave Controller

1) Parameterization, training and convergence of each slave agent: The parameterization of the neural networks for each slave agent is achieved through various a priori tests. It is already known that, when a neural network is too flexible with high number of hidden layers and nodes, then the obtained function can overfit the input data [23]. When the configuration is inflexible with low number of hidden layers and nodes, then the trained function can give poor generalization on the given state space. To minimize the impact of these problems, we set the training stage for each slave agent to 10^7 TTIs, and the number of active users is randomized once every 1000 TTIs.

Figure 8 illustrates the mean cost functions for each slave agent when the neural networks are configured from one hidden layer and 80 hidden nodes. As expected, the action-value cost functions get lower than that of the value cost functions in all cases since the target values Q_p^T are fixed, while V_p^T depends on the calculated reward ϱ_p at each iteration. The convergence of SA_1 and SA_2 looks similar since both `video_1` and `video_2` classes are requesting relatively the same bitrate levels. The convergence analysis is monitored over the entire training period, where the value cost function of each slave agent $p \in \mathcal{P}$ plays a crucial role. The set of weights $\{\theta_{p,1}, \theta_{p,2}, \theta_{p,3}\}$ is saved each time when a new minimum in the value cost function is discovered. As seen from Fig. 8.a for SA_1 (`video_1`), the value cost function gets minimized at around 5×10^6 number of TTIs in the training stage. In the case of agent SA_2 that represents the `video_2` class (Fig. 8.b), the value cost function is minimized at around 8.5×10^6 TTIs in the training stage. This is explicable since `video_2` needs more time to sweep the range of 165 active learners. When training SA_3 for `video_3`, the minimum value cost function is localized at around 5.3×10^6 TTIs (Fig. 8.c), while for agent SA_4 of `video_4`, a period of 7×10^6 TTIs is needed to minimize the value cost function, as illustrated in Fig. 8.d.

2) Testing the Slave Agents: In the exploitation stage, we compare for each video class the performance of the obtained RL-based schedulers (slave agents) with the static approaches, in terms of BF, EXP and OPLF. Figure 9a analyses the performance of the considered strategies for each video class in terms of the normalized number of TTIs when all mobile learners respect the GBR requirements. In the case of `video_1` and `video_2` services, this metric shows similar results. However, when analyzing the performance of `video_3` and `video_4`, SA_3 and SA_4 follow the performance of BF and OPLF schedulers and provide gains higher than 10% when compared to EXP scheduling rule. The same trend can be observed in Fig. 9b where the 5^{th} percentile of user throughput is monitored. Gains higher than 3% can be obtained when employing the RL-based schedulers over the static EXP rule for `video_3` and `video_4` services.

In figures 9c and 9d, the performance of FDP schedulers in terms of delay is presented. As expected, the EXP scheduling rule provides the highest amount of TTIs when the delay objective is met by all learners (Fig. 9c). This aspect becomes more obvious in the cases of `video_3` and `video_4` services, where the EXP rule gets gains higher than 30% when compared to BF scheduler. The RL-based slave agents get nearly similar performance reported to EXP static rule, and when the 95^{th} delay percentile is analysed (Fig. 9d), the slave agents indicate gains higher than 25% when matched against BF for `video_3` and `video_4` services.

In terms of the PLR performance, all candidates provide close numerical results for `video_2` services when we monitor the normalized number of TTI when all mobile learners meet the PLR requirements (Fig. 9e). In case of `video_1`,

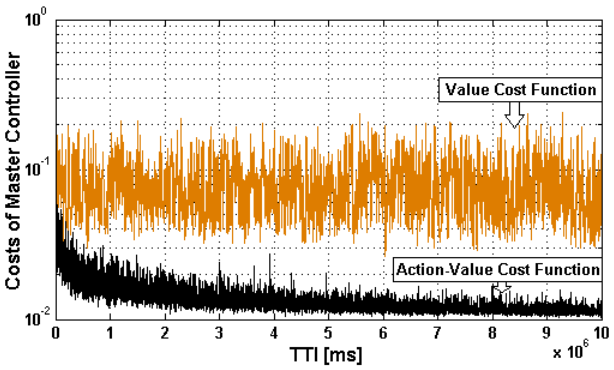


Fig. 10: Training stage of the master controller

the slave agent SA_1 and EXP scheduling rule act best showing gains of 11% compared to BF. The gains get even higher to about 35% when increasing the bit rates of video classes such as `video_3` and `video_4`. In Fig. 9f, we analyse the performance of FDP schedulers in terms of 95^{th} PLR percentile for each video class. The EXP rule and the slave agents afferent to classes `video_1` and `video_2` perform best. For `video_3` and `video_4`, RL-based approaches and OPLF scheduling rule show the best results with gains of 36% compared to EXP and BF scheduling strategies.

By analysing the performance of the slave controller, the following conclusion can be drawn: *a)* the slave agents provides nearly or even better performance compared to the static scheduling scheme that gives the best results for each objective; *b)* the advantage of using the RL approach to select the best scheduling rules becomes more obvious when increasing the data rates of video services; *c)* the STD levels induced by RL-based approaches are not much higher than that of other static FDP scheduling schemes, making the trained slave agents stable options to be employed in real practice.

C. Performance of Slave and Master Controllers

1) Training the master controller while exploiting the decisions of the slave controller: Figure 10 shows the training process of the master controller. In this case, the prioritization sequences are learnt while exploiting the decisions given by the slave controller. The master agents keep the same structure of one hidden layer and 80 hidden nodes. For the neural network approximating the value function $V(\bar{\mathbf{s}};\Theta)$, we use one hidden layer with 200 hidden nodes. This is necessary since this critic neural network takes as input data the entire state $\bar{\mathbf{s}} \in \bar{\mathcal{S}}$ that comprises the observations gathered from all video classes. Because of this aspect, the fluctuations of the value cost function during a training period of 10^7 TTIs are higher when compared to the case of a single slave agent. However, as seen in Fig. 10, the costs of master agents get stabilized over time since the target values are fixed to some specific levels given by (26). By monitoring the value cost function, the agents' weights $\{\Theta_1, \Theta_2, \Theta_3, \Theta_4\}$ are saved each time when this function gets minimized and $\Delta(\Theta) \geq 0$, meaning that, the negative experiences are ignored.

2) Testing the master and slave controllers: The performance of the HiMARL framework is presented in Fig. 11 under the traffic-mix conditions. In this case, both master and slave agents are exploited and their performance is analysed by using the comparison with the state-of-the-art and priority-based schedulers, such as RADS and FLS.

In Fig. 11a we analyse the normalized number of TTIs when the GBR requirements are respected by all mobile learners in each video class. For the first prioritized services `video_1` and `video_2`, HiMARL and FLS act best with gains of 19% and 5%, respectively. In the cases of `video_3` and `video_4`, HiMARL is the best option by offering gains of about 45% and 15% when compared to FLS and RADS, respectively. Figure 11b plots the 5^{th} user throughput percentile of each video class. Similar performance of the employed schedulers can be observed for `video_1` and `video_2` services. However, HiMARL shows its superiority compared to FLS and RADS when analysing the performance of `video_3` and `video_4` services. Compared to FLS, HiMARL provides 42% and 19% more throughput when learners request `video_3` and `video_4` content, respectively.

In terms of delay, we represent in Fig. 11c the normalized number of TTIs when the delay objective is met by all mobile learners in each video class, while in Fig. 11d, the performance of the involved schedulers is presented in terms of the 95^{th} delay percentile at the logarithmic scale. As seen in Fig. 11c, similar numerical results are obtained by all schedulers for the `video_1` class. From Fig. 11d, it can be seen that HiMARL can perform slightly better for the same video class. The same trend can be observed for `video_2` service, with the amendment that, the FLS scheduler can get lower delay percentiles. When delivering `video_3` and `video_4` content, HiMARL is much faster with gains higher than 30% and 13% when compared to FLS and RADS approaches, respectively (Fig. 11c). The same advantages can be observed in Fig. 11d, where HiMARL can deliver much faster `video_3` and `video_4` services to learners with unfavorable channel conditions.

In Fig. 11e, the HiMARL framework obtains the highest amount of TTIs when the PLR requirements are respected for all video classes. In particular, similar to FLS scheme, HiMARL gets gains higher than 40% and 5% for `video_1` and `video_2`, respectively, when compared to RADS scheduler. For the rest of services, HiMARL is the best option being able to get more than 33% and 13% of time when the PLR requirements are respected, in terms of the `video_3` and `video_4` services, respectively. In conjunction with Fig. 11f, it can be observed that FLS obtains lower 95^{th} PLR percentile compared to HiMARL for `video_2` while similar performance between the two is shown in Fig. 11e. It can be concluded that FLS aims to over-provision the mobile learners requesting `video_2` service. For mobile learners requesting content from `video_3` and `video_4` classes, HiMARL is the best option being capable to reduce the loss rates by more than 70% and 30%, respectively, as indicated by Fig. 11f.

Based on the numerical results obtained by comparing the performance of HiMARL with other scheduling candidates, the following conclusion can be formulated: *a)* the proposed HiMARL framework can increase the QoS provisioning of `video_3` and `video_4` services without penalizing the learners receiving educational content from `video_1` and `video_2`; *b)* the FLS strategy is over-provisioning the first prioritized classes in the detriment of `video_3` and

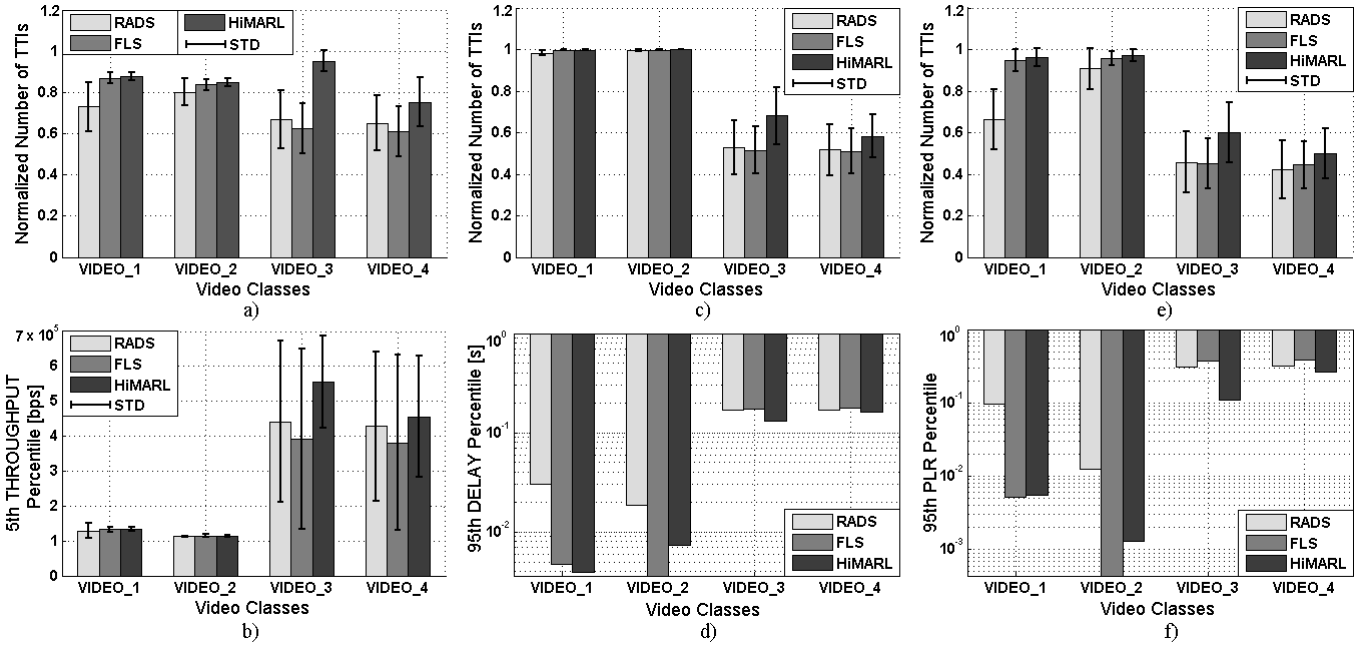


Fig. 11: HiMARL framework performance in terms of $\mu_p(\mathbf{m}_p)$ and $\sigma_p(\mathbf{m}_p)$, where metrics $\mathbf{m}_p$ are: a) normalized number of TTIs when all learners meet GBR requirements; b) 5^{th} throughput percentile; c) normalized number of TTI when all learners respect the delay requirements; d) 95^{th} delay percentile; e) normalized number of TTIs when all learners meet the PLR requirements; f) 95^{th} PLR percentile.

`video_4` services; *c)* RADS is a fair option among all video services, but since the prioritization order of video classes is not considered at the TDP level, then this scheme is unable to differentiate between the current QoS needs of each class.

D. Bridging the Subjective and Objective Assessments

According to the subjective assessments conducted in Section III, students can learn in general regardless of the video quality. The only statistical difference can be observed in the case of slideshow content, where this service must be differentiated between high and low quality, as considered by `video_1` and `video_2` classes, respectively. However, these experiments were conducted per each student in particular and under the presumption of the perfect networking conditions. Thus, the challenge is to address a more realistic scenario where groups of students can access educational content simultaneously, while taking into account the variability of the network environment. In this case, an acceptable QoS provisioning must be ensured by the network scheduler in special, in order to provide to students similar learning achievement levels as obtained in figures 3 and 4.

The proposed hierarchical meta-scheduler based on reinforcement learning addresses the challenges of scheduling multiple mobile learners accessing different educational content at the same time, while considering the variable nature of wireless and networking conditions. The aim is to maximize the number of mobile learners that can get educational video content with enhanced QoS provisioning and without sacrificing their learning achievement level. As part of the proposed HiMARL framework, the slave controller is designed to assure an optimal resource allocation policy that can maximize the QoS revenue for each video class. Figure 12a shows that the slave agents can act in general better than other FDP schedulers when the multi-objective QoS provisioning is monitored separately for each class. By exploiting the decisions given by the slave agents and training the master controller to decide the prioritization sequence to be followed each time, the proposed framework outperforms other candidates in terms of the multi-objective QoS provisioning for each service class. As seen in Fig. 12b, HiMARL ensures a better QoS provisioning in terms of GBR, PLR and delay when delivering the slideshow content at low and high quality (`video_1` and `video_2`). When scheduling mobile learners requesting the animation content at low quality (`video_3`), HiMARL obtains a gain of about 35% compared to RADS and FLS. In the case of screeencast content (`video_4`), HiMARL outperforms other candidates by more than 17% of time when all QoS objectives are met.

Fig. 12c illustrates the normalized number of TTIs when all QoS objectives are met in all classes simultaneously as a function of aggregate number of mobile learners. It can be observed a relatively similar performance for all involved schedulers when $L \leq 18$ since the radio resources are enough to cope with low number of aggregate learners and hence, the optimization of scheduling and resource allocation does not bring any significant performance gains. When the network is saturated and $L \geq 46$, the number of TTIs when all QoS objectives are simultaneous met is close to zero for all schedulers. In this case, HiMARL and RADS share the available bandwidth between all classes with a certain inter-class fairness, imposed by the standardized prioritized order of $\mathcal{P}$ [31], as shown in Fig. 11f. The FLS scheduler over-provisions the high priority classes e.g. `video_1` and `video_2`. However, for heterogeneous learners in $L \in [24, 42]$ range, HiMARL shows significant performance gains when compared to FLS and RADS. Since the level of heterogeneous QoS provisioning of HiMARL for 34 learners is equivalent with the level obtained by RADS and FLS schedulers for 24 learners, we conclude that HiMARL can accommodate up to 41% more learners.

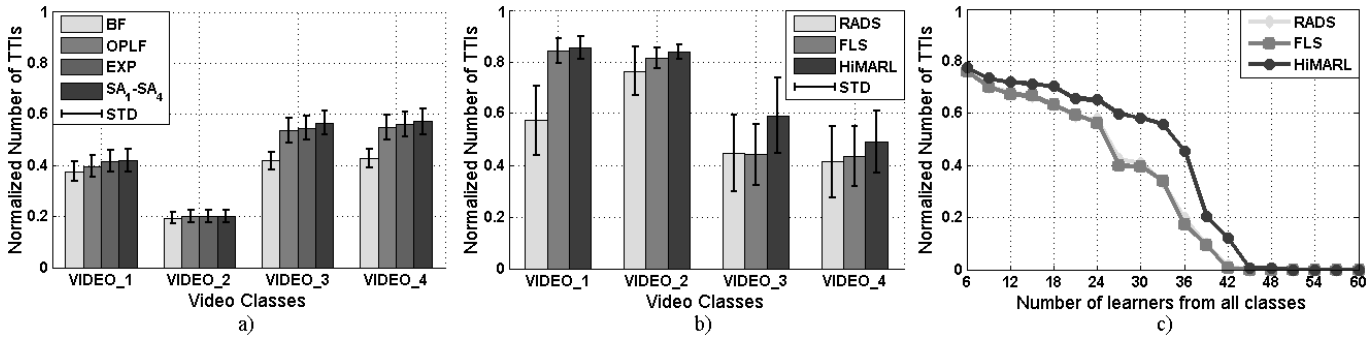


Fig. 12: a) Performance of slave agents in terms of $\mu_p(\mathbf{m}_p)$ and $\sigma_p(\mathbf{m}_p)$, where $\mathbf{m}_p$ is the normalized number of TTIs when all objectives are met; b) Performance of the HiMARL framework in terms of $\mu_p(\mathbf{m}_p)$ and $\sigma_p(\mathbf{m}_p)$, where $\mathbf{m}_p$ is the normalized number of TTIs when all objectives are met; c) Performance of the HiMARL framework in terms of $\mu(\mathbf{m})$, where $\mathbf{m}$ is the normalized number of TTIs per total number of mobile learners when all objectives are met.

VII. Conclusion

This paper proposes a novel machine learning-based resource allocation solution for the OFDMA-based networks that makes use of a novel Hierarchical decision-making process based on Multi-Agent Reinforcement Learning (HiMARL) to enable access to video content to an increased number of mobile learners without sacrificing their learning achievements. The proposed HiMARL framework employs a master controller to learn the most suitable prioritization sequence of educational video classes and a slave controller that approximates the best scheduling rules to be employed by each class for the resource allocation. The performance evaluation results show that the HiMARL framework can accommodate up to 41% more mobile learners for the same quality achievements as compared to other state-of-the-art schedulers.

References

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (vni), complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018.

[2] N. Saeed, A. Bader, T. Y. Al-Naffouri, and M.-S. Alouini, "When wireless communication responds to covid-19: Combating the pandemic and saving the economy," *Frontiers in Communications and Networks*, vol. 1, p. 3, 2020.

[3] ITU, "Special emergency session of the broadband commission pushes for action to extend internet access and boost capacity to fight covid-19," 2020. [Online]. Available: https://www.itu.int/en/mediacentre/Pages/PR05-2020-Broadband-Commission-emergency-session-internet-COVID-19.aspx

[4] A. Sepúlveda, "The Digital Transformation of Education: Connecting Schools,Empowering Learners," *Broadband Commission for Sustainable Development's Working Group on School Connectivity, International Telecommunication Union*, 2020.

[5] I.-S. Comșa, G.-M. Muntean, and R. Trestian, "An Innovative Machine-Learning-Based Scheduling Solution for Improving Live UHD Video Streaming Quality in Highly Dynamic Network Environments," *in IEEE Transactions on Broadcasting*, vol. 1, no. 1, pp. 1 – 13, April 2020.

[6] C. M. Toquero, "Challenges and Opportunities for Higher Education amid the COVID-19 Pandemic: The Philippine Context," *Pedagogical Research*, vol. 5, no. 4, pp. 1 – 5, Apr. 2020.

[7] S. Dhawan, "Online learning: A panacea in the time of covid-19 crisis," *Journal of Educational Tech. Systems*, vol. 49, no. 1, pp. 5–22, 2020.

[8] R. Ubell, "How online learning kept higher ed open during the coronavirus crisis," May 2020. [Online]. Available: https://spectrum.ieee.org/tech-talk/at-work/education/how-online-learning-kept-higher-ed-open-during-the-coronavirus-crisis

[9] A. Khattar, P. R. Jain, and S. M. K. Quadri, "Effects of the disastrous pandemic covid 19 on learning styles, activities and mental health of young indian students - a machine learning approach," in *Intl. Conf. on Intelligent Computing & Control Syst. (ICICCS)*, 2020, pp. 1190–1195.

[10] A. Sönmez, L. Göçmez, D. Uygun, and M. Ataizi, "A review of current studies of mobile learning," *Journal of Educational Technology and Online Learning*, vol. 1, pp. 12 – 27, 2018.

[11] M. Al-Emran, H. M. Elsherif, and K. Shaalan, "Investigating attitudes towards the use of mobile learning in higher education," *Computers in Human Behavior*, vol. 56, pp. 93 – 102, 2016.

[12] A. Naciri, M. A. Baba, A. Achbani, and A. Kharbach, "Mobile Learning in Higher Education: Unavoidable Alternative during COVID-19," *Aquademia*, vol. 4, no. 1, pp. 1 – 2, Apr. 2020.

[13] J. Romero-Rodríguez, I. Aznar-Díaz, F. Hinojo-Lucena, and G. Gómez-García, "Mobile learning in higher education: Structural equation model for good teaching practices," *IEEE Access*, vol. 8, pp. 91 761–91 769, 2020.

[14] M. Wyres and N. Taylor, "Covid-19: using simulation and technology-enhanced learning to negotiate and adapt to the ongoing challenges in UK healthcare education," *in BMJ Simulation and Technology Enhanced Learning*, vol. 6, no. 6, pp. 317–319, Nov. 2020.

[15] G.-M. Muntean, "Efficient delivery of multimedia streams over broadband networks using qoas," *IEEE Transactions on Broadcasting*, vol. 52, no. 2, pp. 230–235, 2006.

[16] L. Zou, T. Bi, and G.-M. Muntean, "A dash-based adaptive multiple sensorial content delivery solution for improved user quality of experience," *IEEE Access*, vol. 7, pp. 89 172–89 187, 2019.

[17] A. N. Moldovan and C. H. Muntean, "DQAMLearn: Device and QoE-Aware Adaptive Multimedia Mobile Learning Framework," *IEEE Transactions on Broadcasting*, vol. 99, pp. 1–16, 2020.

[18] Z. Yuan, G. Ghinea, and G.-M. Muntean, "Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 104–117, 2015.

[19] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360 video streaming: Solutions, challenges and opportunities," *IEEE Communications Surveys Tutorials*, pp. 1–24, 2020.

[20] G.-M. Muntean and N. Cranley, "Resource efficient quality-oriented wireless broadcasting of adaptive multimedia content," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 362–368, 2007.

[21] L. Zou, R. Trestian, and G.-M. Muntean, "Doas: Device-oriented adaptive multimedia scheme for 3gpp lte systems," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 2180–2184.

[22] Z. Yuan and G.-M. Muntean, "A prioritized adaptive scheme for multimedia services over IEEE 802.11 WLANs," *IEEE Transactions on Network and Service Management*, vol. 10, no. 4, pp. 340–355, 2013.

[23] I.-S. Comșa, R. Trestian, G. M. Muntean, and G. Ghinea, "5MART: a 5G SMART Scheduling Framework for Optimizing QoS Through Reinforcement Learning," *in IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1110–1124, Dec. 2019.

[24] A. Molnar, "Content type and perceived multimedia quality in mobile learning." *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21 613–21 627, 2017.

[25] R. E. Mayer, *Multimedia Learning*. Cambridge Univ. Press, April 2001. [Online]. Available: https://www.amazon.com/gp/product/0521787491

[26] A. Molnar and C. H. Muntean, "Assessing learning achievements when reducing mobile video quality." *J. Univers. Comput. Sci.*, vol. 21, no. 7, pp. 959–975, 2015.

[27] ITU-T, "Recommendation ITU-T P.910: Subjective video quality assessment methods for multimedia applications," International Telecommunication Union, 2008.

[28] D. W. Zimmerman, "Comparative power of student t test and mann-whitney u test for unequal sample sizes and variances," *The Journal of Experimental Education*, vol. 55, no. 3, pp. 171–174, 1987.
[29] A. Molnar and C. H. Muntean, "Cost-oriented adaptive multimedia delivery," *IEEE Trans. Broadcasting*, vol. 59, no. 3, pp. 484–499, 2013.
[30] I.-S. Comșa, S. Zhang, M. Aydin, P. Kuonen, L. Yao, R. Trestian, and G. Ghinea, "Towards 5G: A Reinforcement Learning-based Scheduling Solution for Data Traffic Management," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1661–1675, Aug. 2018.
[31] 3GPP, "Technical Specification Group Services and System Aspects; Policy and charging control architecture Release 12, v.12.2.0," 3GPP, Tech. Rep., 2013.
[32] F. Avocanh, M. Abdennebi, and J. Ben-Othman, "An Enhanced Two Level Scheduler to Increase Multimedia Services Performance in LTE Networks," in *IEEE Intl. Conf. on Communications (ICC)*, June 2014, pp. 2351–2356.
[33] G. Monghal, D. Laselva, P.-H. Michaelsen, and J. Wigard, "Dynamic Packet Scheduling for Traffic Mixes of Best Effort and VoIP Users in E-UTRAN Downlink," in *IEEE Vehicular Technology Conference (VTC-Spring)*, May 2010, pp. 1 – 5.
[34] G. Piro, L. Grieco, G. Boggia, R. Fortuna, and P. Camarda, "Two-Level Downlink Scheduling for Real-Time Multimedia Services in LTE Networks," *in IEEE Trans. on Multimedia*, vol. 13, p. 1052–1065, 2011.
[35] W. Chung, C. J. Chang, and L. Wang, "An Intelligent Priority Resource Allocation Scheme for LTE-A Downlink Systems," *in IEEE Wireless Communications Letters*, vol. 1, no. 3, pp. 241 – 244, 2012.
[36] K. Wang, X. Li, H. Ji, and X. Zhang, "Heterogeneous Traffic Scheduling in Downlink High Speed Railway LTE Systems," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 1452–1457.
[37] M. Lundevall, B. Olin, J. Olsson, N. Wiberg, S. Wanstedt, J. Eriksson, and F. Eng, "Streaming Applications Over HSDPA in Mixed Service scenarios," in *IEEE Vehicular Technology Conference VTC2004-Fall*, vol. 1, April 2005, pp. 841 – 845.
[38] B. Sadiq, R. Madan, and A. Sampath, "Downlink Scheduling for Multiclass Traffic in LTE," *in EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 14, pp. 1–18, 2009.
[39] N. Khan, M. Martini, Z. Bharucha, and G. Auer, "Opportunistic Packet Loss Fair Scheduling for Delay-Sensitive Applications over LTE Systems," in *IEEE Wireless Communications and Networking Conference*, vol. 1, April 2012, pp. 1456 – 1461.
[40] I.-S. Comșa, *Sustainable Scheduling Policies for Radio Access Networks Based on LTE Technology*. University of Bedfordshire, U.K., 2014.
[41] C. Szepesvári, *Algorithms for Reinforcement Learning: Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan and Claypool Publishers, 2010.
[42] L. Busoniu, R. Babuska, and B. D. Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156 – 172, February 2008.
[43] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.
[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rus, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level Control through Deep Reinforcement Learning," *Nature*, vol. 518, p. 529–533, Feb. 2015.
[45] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework," *in IEEE Transactions on Vehicular Networks*, vol. 60, no. 2, pp. 498 – 513, 2011.

Ioan-Sorin Comșa is a Data Scientist at the Swiss Distance University of Applied Sciences. He received his PhD degree from the Institute for Research in Applicable Computing, University of Bedfordshire UK, in June 2015. He was also a PhD Researcher with the Institute of Complex Systems, University of Applied Sciences of Western Switzerland and he worked as a Research Engineer at CEA-LETI Grenoble France. Since 2017, he was a Research Assistant at Brunel University London. His research interests include intelligent radio resource and QoS management, reinforcement learning, data mining, distributed and parallel computing, adaptive multimedia/mulsemedia delivery and eLearning.

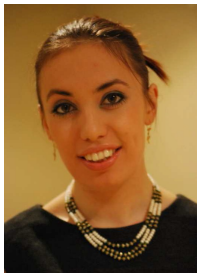
Andreea Molnar is a Senior Lecturer at Swinburne University of Technology, Melbourne, Australia. She received a PhD on Technology Enhanced Learning from National College of Ireland. Her research interest include video based learning, serious games and virtual reality. She is a Senior Editor for the Information Technology People and in the Editorial Board for the International Journal of Game-based Learning.

Irina Tal is an Assistant Professor with School of Computing, Dublin City University in Ireland, Academic Lead of the MSc in Blockchain and member of LERO. She received her Ph.D. degree from the School of Electronic Engineering, Dublin City University, Ireland. Her research interests include technology enhanced learning, vehicular ad-hoc networks, smart cities and cyber security. She is the Lead Principal Investigator on the SFI funded project PRIVATT. She published in prestigious international conferences and journals.

Per Bergamin is Professor for Didactics in Distance Education and E-Learning at the Swiss Distance University of Applied Sciences (FFHS). Since 2006 he acts as the Director of the Institute for Research in Open-, Distance- and eLearning (IFeL) and from 2016 on he holds also the UNESCO Chair on personalised and adaptive Distance Education. In 2020 he is also appointed as extraordinary professor at the Faculty of Education of North-West University (NWU, ZA). His research activities focus on self-regulated and technology-based personalized and adaptive learning. Central aspects are learning analytics, instructional design, usability and application implementation. As a researcher he cooperates with or leads different national and international projects and contributes to different Swiss advisory boards for E-Learning development. As a teacher he covers the topics of Educational Psychology and E-Didactics. Furthermore he was founder and president of the Executive Board of a Company for E-Business and BI as well as Learning Applications, which he sold in 2016.

Gabriel-Miro Muntean (M'04–SM'17) is a Professor with the School of Electronic Engineering, Dublin City Univ. (DCU), Ireland, and co-Director of the DCU Performance Engineering Lab. He has published over 450 papers in top international journals and conferences, authored 4 books and 22 book chapters, and edited 7 other 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 top international journals, conferences, and funding agencies.

Cristina Hava Muntean received her Ph.D. degree from Dublin City University, Ireland in 2005. She is a Senior Lecturer with the School of Computing, National College of Ireland. She has been involved in various research activities over the past 15 years fostering and promoting research, leading research projects, supervising PhD and MSc students and publishing over 80 publications in international peer-reviewed books, journals, and conferences. Her main research areas are adaptive multimedia, adaptive and personalized learning and user quality of experience.



Ramona Trestian is a Senior Lecturer with the Design Engineering and Mathematics Dept., Middlesex Univ., London, UK. She received her Ph.D. degree from Dublin City Univ., Ireland in 2012. She published in prestigious international conferences and journals and has five edited books. Her research interests include mobile and wireless communications, quality of experience, multimedia streaming, handover and network selection strategies, digital twin modelling, etc. She is an Associate Editor of the IEEE Communications Surveys and Tutorials.