Improved Quality of Online Education Using Prioritized Multi-Agent Reinforcement Learning for Video Traffic Scheduling

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

***Abstract*—The recent global pandemic has transformed the way education is delivered, increasing the importance of video-based online learning. However, this puts a significant pressure on the underlying communication networks and the limited available bandwidth needs to be intelligently allocated to support a much higher transmission load, including video-based services. In this context, this paper proposes a Machine Learning (ML)-based solution that dynamically prioritizes content viewers with heterogeneous video services to increase their Quality of Service (QoS) and perceived Quality of Experience (QoE). The proposed approach makes use of the novel Prioritized Multi-Agent Reinforcement Learning solution (PriMARL) to decide the prioritization order of the video-based services based on networking conditions. However, the performance in terms of QoS and QoE provisioning to learners with different profiles and networking conditions depends on the type of scheduler employed in the frequency domain to conduct the scheduling and the radio resource allocation. To decide the best approach to be followed, we employ the proposed PriMARL solution with different types of scheduling rules and compare them with other state-of-the-art solutions in terms of throughput, delay, packet loss, Peak Signal-to-Noise Ratio (PSNR), and Mean Opinion Score (MOS) for different traffic loads and characteristics. We show that the proposed solution achieves the best user QoE results.**

***Index Terms*—Machine learning, multi-agent reinforcement learning, video traffic prioritization, QoE, online education.**

Manuscript received 9 November 2022; revised 26 January 2023; accepted 9 February 2023. Date of publication 16 March 2023; date of current version 7 June 2023. *(Corresponding author: Gabriel-Miro Muntean.)*

Ioan-Sorin Comșa and Christof Imhof are with the Institute for Research in Open-, Distance- and eLearning, Swiss Distance University of Applied Sciences, 3900 Brig, Switzerland (e-mail: ioan-sorin.comsa@ffhs.ch; christof.imhof@ffhs.ch).

Andreea Molnar is with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia, and also with the Institute for Advance Studies, Technical University of Munich, 85748 Garching, Germany.

Irina Tal and Gabriel-Miro Muntean are with the Performance Engineering Laboratory, Dublin City University, Dublin 9, D09 DXA0 Ireland (e-mail: irina.tal@dcu.ie; gabriel.muntean@dcu.ie).

Per Bergamin is with the Institute for Research in Open-, Distance- and eLearning, Swiss Distance University of Applied Sciences, 3900 Brig, Switzerland, and also with the Faculty of Education, North-West University, Potchefstroom 2531, South Africa (e-mail: per.bergamin@ffhs.ch).

Cristina Hava Muntean is with the School of Computing, National College of Ireland, Dublin 1, D01 K6W2 Ireland (e-mail: cristina.muntean@ncirl.ie).

Ramona Trestian is with the Faculty of Science and Technology, Middlesex University London, NW4 4BT London, U.K.

Digital Object Identifier 10.1109/TBC.2023.3246815

I. Introduction

BROADBAND connectivity plays a central role in mitigating the economic aftermath of the pandemic and boosting the digital access and inclusiveness of different sectors [1]. One such sector of utmost importance is remote education and eLearning, which all regions of the world must have access to [2]. COVID-19 containment measures forced actors of the educational sector to remotely deliver large amounts of media content across the existing broadband infrastructure. Prior to the global pandemic, educational institutions were slowly moving towards a blended learning approach which combines the traditional physical classroom teaching with the adoption of various Information and Communication Technology (ICT)-based tools and solutions to improve the educational experience [3]. However, the global pandemic has accelerated the digital transformation of educational institutions by forcing the teaching-learning process to move to *'online only'*. In this context, instructors rely on any form of video content (e.g., live video streaming, video on demand, etc.) as well as text and graphics, to improve the teaching-learning process within the online learning environment. Previous studies [4] have shown that the integration of instructional videos within the educational content can increase the effectiveness of online learning. However, moving from the optional adoption of ICT-based tools within the educational domain to a compulsory one, including video-based learning, does not come without challenges.

One of the existing challenges that was worsened by the pandemic is the issue of digital inequalities. The factors that contribute to these inequalities are [5]: (1) digital literacy; (2) access to hardware and/or software; (3) usage autonomy; and (4) social factors, such as peer interactions. Additionally, when it comes to video-based learning, there are several factors that impact learners' Quality of Experience (QoE), including the type of device (e.g., smartphone, laptop, desktop, etc.) and the quality of the Internet connectivity. To be able to accommodate an appropriate level of eLearning content for mobile learners, stable broadband connections are strongly demanded. To this end, network operators are pressured to ensure high levels of Quality of Service (QoS) and QoE while exchanging larger amounts of educational media content among an increasing number

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

of mobile/online learners over the existing radio access networks.

Enabling good QoS provisioning over the wireless interface is challenging. A limited frequency spectrum must be allocated by a scheduling entity to increase the number of users requesting different traffic types and experiencing a variety of network conditions [6]. In remote education, this aspect is even more of a challenge since a proper prioritization of the delivered services is needed to deal with different learner profiles, dynamic wireless conditions, device types, and content characteristics with heterogeneous QoS requirements [7]. Therefore, the focus of this paper is on packet scheduler and intelligent prioritization of eLearning content for mobile learners. To provide high QoS, we employ a solution based on Prioritized Multi-Agent Reinforcement Learning (PriMARL) [8] to allocate the limited frequency spectrum over an increased number of mobile learners accessing the radio interface. However, enabling high QoS provisioning does not guarantee acceptable QoE when scheduling video services with different degrees of heterogeneity in terms of data rates and QoS requirements. Therefore, the focus of PriMARL would be to maximise both QoS and QoE provisioning for learners experiencing heterogeneous video services in eLearning.

In the literature, multi-agent reinforcement learning is used to deal with user association and resource allocation in heterogeneous cellular and millimeter wave networks [9], [10]. In our previous work [7], we considered the prioritization and scheduling aspects of educational content over the broadband networks and proposed a Hierarchical MARL (HiMARL) model based on a source-sync approach, where the source controller prioritizes video classes in the time domain and the sync controller performs the scheduling and resource allocation in the frequency domain. This method is highly efficient to deliver the requested heterogeneous video services in terms of QoS compared to other state-of-the-art approaches. However, there is no evaluation of the proposed scheduling technique in terms of QoE.

A. Addressed Use Case Scenario

According to a study conducted by Campbell [11], one of the most important issues in enabling eLearning over mobile technology (mobile learning) is the network speed and reliability. In this context, the use case scenario illustrated in Figure 1 is considered. Four types of mobile users access educational video services from a cloud mobile learning server via a 5G gNodeB base station. The mobile users are located in different geographical locations, use diverse device types (e.g., smartphones, laptops, tablets, VR gear, etc.), and have various network connectivity characteristics (e.g., poor, medium, or good connectivity). In this scenario, the network scheduler located at the level of the 5G gNodeB base station is responsible for allocation of the available radio resources to all users and maximizing the QoS parameters for each delivered video service, given the channel conditions, traffic types and characteristics, device resolutions, and prioritization policies. However, as noted in [7], the quality of user experience is important for the learning performance. Therefore, in this

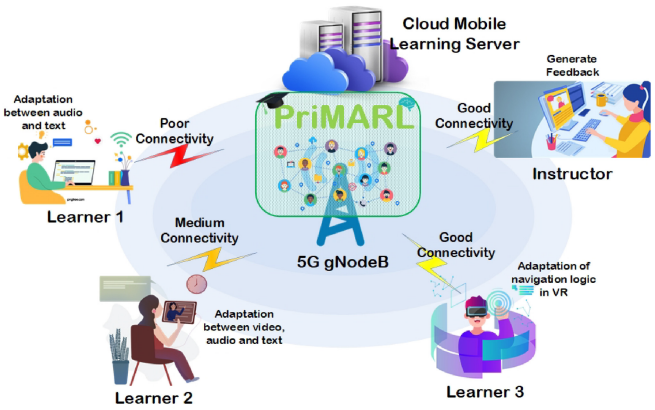

Fig. 1. Use Case Scenario.

paper, we propose PriMARL, an ML-based decision-making framework that aims to increase the time and number of users (learners, instructors) experiencing high QoE levels when delivering a range of four video services.

B. Paper Contributions

The proposed PriMARL framework for downlink scheduling systems eliminates the need for a source-sync approach as employed in the previous work (HiMARL) and improves learner QoE when delivering heterogeneous educational video in different traffic load conditions. In contrast to [7], the contributions of this paper are as follows.

a) Prioritization-Driven Scheduler: The proposed approach focuses on service prioritization. It provides a low complexity solution to the proposed optimization problem that decides in each Transmission Time Interval (TTI) the prioritization order of video classes with different QoS profiles in the time domain and considers particular scheduling rules in the frequency domain, i.e., Barrier Function (BF), Exponential (EXP) and Opportunistic Packet Loss Fair (OPLF) [6].

b) PriMARL-based Decision-Making: Three different PriMARL solutions which employ various scheduling rules, i.e., PriMARL-BF, PriMARL-EXP and PriMARL-OPLF to maximise QoS and QoE are designed, trained, and tested. The functional framework allows training and testing of PriMARL policies under the same network and traffic conditions, ensuring high accuracy of comparison and conclusions. Compared to HiMARL [7], the proposed solutions provide improved user perceived QoE when delivering video services with different traffic loads (low, medium, and high).

c) Higher Number of Learners Experiencing Video Content at Excellent QoE Level: Unlike previous work, this research focuses on improving user QoE, estimated in terms of Peak Signal-to-Noise Ratio (PSNR) and Mean Opinion Score (MOS). For instance, the proposed PriMARL-EXP solution increases the number of learners experiencing excellent QoE levels for the considered video classes at different traffic load settings.

The remainder of this paper is organized as follows: In Section II, we discuss the related work carried out in this area. Section III introduces the system model, and in Section IV, we

describe the proposed PriMARL-based solution. In Section V, we present an analysis of obtained results and Section VI serves as the conclusion of our paper.

II. Related Works

Recently, an increasing number of solutions that make use of Machine Learning (ML) and other Artificial Intelligence (AI) techniques have started gaining momentum in various fields, mainly due to the global pandemic that accelerated the digital transformation. Different ML-based approaches are proposed in the literature to build intelligent systems that identify patterns and behaviour in historical data and learn from it without relying on rules-based systems.

The concept of *Multimedia Intelligence* is introduced by Zhu et al. [12], representing the convergence of multimedia and AI. A bidirectional link is formed between multimedia and AI, that enables them to enhance each other. Consequently, on one side, multimedia enriches the varieties of applications for AI through explainability. On the other side, AI boosts the inferrability of multimedia through reasoning.

Deep Reinforcement Learning (DRL) has been used by Cui et al. [13] to propose TCLiVi, a transmission control in live video streaming solutions. TCLivi jointly adjusts the streaming parameters (e.g., video bitrate, target buffer size) in order to improve the QoE for live video streaming. The performance evaluation results show that TCLiVi outperforms other solutions from the literature in terms of QoE score with an increase of 40.84%. DRL has also been used by Mao et al. [14] to propose Pensieve, an intelligent system that generates adaptive bitrate (ABR) algorithms for Video on Demand (VoD) scenarios. Pensieve will automatically learn the adaptive bitrate algorithms that adapt to a wide range of dynamic network conditions and QoE metrics.

Tan et al. [15] investigate the use of game theory to enable dynamic adaptive bitrate streaming in multi-client over Named Data Networking (NDN). A client-side game theory-based distributed ABR algorithm for NDN is proposed to optimize the overall QoE of multiple clients and guarantee fairness. The performance evaluation results demonstrate the effectiveness of the proposed solution in terms of overall QoE, fairness, and bandwidth resource utilization. Looking at maximizing user capacity for an auto-scaling VoD system, Chang and Chan [16] propose AVARDO, an auto-scaling Video Allocation and Request Distribution Optimization solution. The proposed solution seeks to maximize the user capacity at each auto-scaling level and formulate the optimization problem as a multi-objective mixed-integer linear programming problem. The performance evaluation results show that the proposed AVARDO solution is close to the optimum.

Random Forest (RF) classifier is used by Chandrasekhar et al. [17] for real time video scheduling over LTE networks. The proposed solution detects the service type of different flows as well as the video player status for users with HTTP Adaptive Streaming (HAS) flows. The output of the RF classifier is used for prioritizing scheduling of the HAS users. The proposed solution enhances the video QoE with an acceptable impact on other non-video best effort services. Similarly, an adaptive resource scheduling solution named AdaptSch, based on neural network (NN) and mobile traffic prediction, is proposed by Semov et al. [18]. AdaptSch makes use of an NN architecture with two building blocks, where the first one predicts the future network state, while the second one chooses the optimum scheduling policy to be applied. The proposed solution improves the system performance in terms of packet delay. However, this comes at the cost of overall throughput degradation.

With a focus on radio resource scheduling in the 5G Radio Access Network (RAN), Tseng et al. [19] designed a modularized Deep Deterministic Policy Gradient (DDPG) architecture. Here, DDPG is used to select a radio resource scheduling policy from a pool of 60 combinations of scheduling algorithms as actions. DDPG has been widely adopted to solve optimal control problems in wireless network environments, such as, in case of network slicing for allocating resources among different slices [20], or among different traffic classes [21]. However, Gu et al. [22] argue that due to the very slow convergence of DDPG, it cannot be implemented in real-world 5G systems. Consequently, the authors propose a knowledge-assisted DDPG that reduces its convergence time significantly and achieves better QoS.

Motivated by the fact that reconfigurable wireless networks open up new opportunities for advanced rich multimedia applications, such as online AR/VR gaming, high-quality video streaming, and autonomous vehicles, Mollahasani et al. [23] take a different approach and propose an Actor-Critic learning-based QoS-aware scheduler to overcome the problem of stringent QoS requirements of such applications. The authors adopt two advantage actor-critic models, where the first technique schedules packets by prioritizing their scheduling delay budget, while the second technique considers channel quality, delay budget, and packet type. Performance evaluation results validate the efficiency of the proposed approach.

In addition to the approaches described above, there are also time-efficient schedulers that target multiple QoS objectives at the same time. An example of such a scheduler is the Frame Level Scheduler (FLS) [24] that divides the scheduling problem in two stages: *a)* time-domain, where the users are prioritized based on the approximated quota of data necessary to meet the delay constraints; *b)* frequency-domain, where the prioritized users get radio resources for data transmission in a fair manner according to scheduling rules, such as proportional-fair scheduling. Another efficient example is the Required Activity Detection Scheduler (RADS) [25], where in the time domain, users are prioritized based on a multi-target criterion encompassing fairness, delay, and rate requirement, while in the frequency domain, the pre-selected users are served based on their channel quality. More recently, in [26] the authors proposed the Minimal Delay Violation (MDV) downlink scheduler that considers arrival rates in data queues and the state of each flow in the network in terms of packet loss and delay. When compared to FLS, MDV achieves a maximum gain of about 25% in terms of average system throughput when scheduling users requesting heterogeneous traffic in terms of video, voice, and best effort. In a railway environment, the authors in [27] proposed a New version of RADS (NRADS)

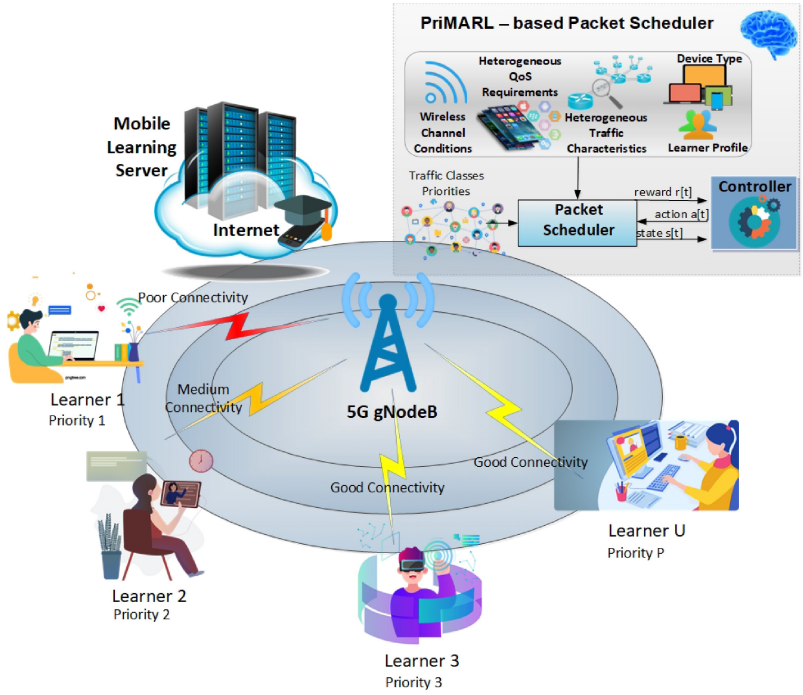

Fig. 2. Proposed System Model.

that allocates the radio resources to mobile users based on the number of correctly received bits at the level of physical layer, channel conditions, and a static and standardized prioritization sequence to be followed when scheduling multiple classes of services. Compared to RADS, NRADS provides a gain of nearly 10% when measuring the overall system throughput.

In summary, a variety of scheduling approaches exists in the literature to deal with prioritization and scheduling of multimedia services. However, most of these approaches are mainly focusing on QoS optimization. Improving user QoE of the provided services, assessed in terms of objective (e.g., PSNR) or subjective (e.g., MOS) metrics, remains uncovered. Despite the amount of research done in these areas, advancements therein would benefit from the performance of our proposed PriMARL-based decision making solution, which focuses on the maximization of PSNR performance for heterogeneous video scheduling given the dynamic user traffic and network conditions. The primary objective of the PriMARL-based prioritization framework is to maximize the QoS revenue for all video content viewers in terms of packet delay, throughput, and packet loss rate (PLR). Then, the second objective would be to carefully select the best rule for the frequency domain-based scheduling that provides the highest amount of viewers with excellent MOS scores.

III. System Model and Problem Statement

The proposed system model is presented in Fig. 2, where mobile/online learners access different types of educational video content from the mobile learner server through the OFDMA interface and scheduling system. Let us define by $\mathcal{P} = \{1, 2, \ldots, P\}$ the set of video services that needs to be prioritized at each TTI, where class 1 requests the highest priority and class P is associated with the lowest priority. Furthermore, we consider by $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_P\}$ the set of active mobile learners distributed over P video classes. Each learner $u \in \mathcal{U}_p$ receives on a mobile device (e.g., tablet, smartphone) educational videos with different QoS constraints or requirements for each class $p \in \mathcal{P}$. By $\mathcal{Q}_p = \{q_{p,n} : n = 1, 2, \ldots, N\}$ we define the set of QoS requirements associated to class $p \in \mathcal{P}$, where n is a type of QoS indicator that can be throughput, delay, or packet loss.

We define Key Performance Indicators (KPI) for the QoS data (i.e., throughput, delay, packet loss) which are measured in each TTI based on observations collected from each user. In multi-class prioritization and scheduling, in a given class $p \in \mathcal{P}$, users' KPIs are constrained by the same set of QoS requirements $\mathcal{Q}_p$ indicated by standards [28]. Therefore, for each QoS type n, class $p \in \mathcal{P}$, and learner $u \in \mathcal{U}_p$, we define the KPI $k_{p,u,n}$ measured at each TTI and monitored to verify if its QoS requirement $q_{p,n}$ is met. By enlarging the dimension of data to the user level for all N QoS indicators, we can further define the learner KPI vector as

$$\mathbf{k}_{p,u} = \left[k_{p,u,n}\right]_{n=1,2,\ldots,N}$$

and the vector of QoS requirements as

$$\mathbf{q}_p = \left[q_{p,n}\right]_{n=1,2,\ldots,N}.$$

Then, the aim is to maximize in each TTI the number of KPI vectors $\mathbf{k}_{p,u}$ respecting the corresponding QoS requirement vector $\mathbf{q}_p$ for as many learners $u \in \mathcal{U}_p$ as possible.

The role of the scheduler from Fig. 2 is to prioritize learners from different video classes $p \in \mathcal{P}$ and allocate the necessary radio resources in the frequency domain at each TTI. Let us suppose that the prioritization sequence, for example,

$$\left[p, 2, \ldots, p-1, p+2, \ldots, 1\right]$$

is decided at TTI t, where learners requesting video service from class $p \in \mathcal{P}$ are scheduled first, followed by learners from class 2, and so on. In the proposed system, the number of video classes from the prioritization sequence that are scheduled in the frequency domain depends on the amount of remaining radio resources. In OFDMA networks, the available bandwidth is divided in B number of equal Resource Blocks (RBs). Let $\mathcal{B} = \{1, 2, \ldots, B\}$ be the set of RBs that are allocated at each TTI, where RB $b \in \mathcal{B}$ is the smallest resource unit. Learners $u \in \mathcal{U}_p$ within video class $p \in \mathcal{P}$ from the prioritized sequence are competing in the frequency domain to get the highest amount of RBs. Then, utility functions are used to rank learners for each RB $b \in \mathcal{B}$ according to their QoS budget [29]. In particular, for each RB $b \in \mathcal{B}$ and learner $u \in \mathcal{U}_p$, an utility function targets specific types of QoS indicators in terms of n and takes as input in each TTI t the measured KPI $k_{p,u,n}$; in most cases, as output, such a function provides a measure of how far each KPI of each class $p \in \mathcal{P}$ is from the QoS requirement $q_{p,n} \in \mathcal{Q}_p$. At the level of each RB $b \in \mathcal{B}$, the learner with the highest utility value is allocated that particular RB. Learners $u \in \mathcal{U}_p$ with higher utility values over the entire bandwidth have higher chances to get more RBs. Let $\Gamma_n(k_{p,u,n}) : \mathbb{R} \rightarrow \mathbb{R}$ be such utility functions that can take different forms depending on the target type of QoS indicator (i.e., throughput, delay, PLR).

A. Optimization Problem

In the proposed optimization problem presented in (1.a), the prioritization of video classes and resource allocation are performed at each TTI $t \in \{1, 2, \ldots, T\}$, subject to constraints (1.b)-(1.e), where T represents the number of TTIs of a given scheduling session.

$$\max_{x,y} \sum_{p\in\mathcal{P}} \sum_{u\in\mathcal{U}_p} \sum_{b\in\mathcal{B}} x_{p,u}(t) \cdot y_{u,b}(t) \cdot \Gamma_n\left[k_{p,u,n}(t)\right] \cdot \lambda_{u,b}(t),$$

s.t. (1.a)

$$\sum_{u} y_{u,b}(t) \leq 1, \quad b = 1, \ldots, B, \tag{1.b}$$

$$\sum_{p} x_{p,u}(t) = 1, \quad u = u_1, \ldots, u_{U_p}, p = 1, \ldots, P, \tag{1.c}$$

$$\sum_{p^*} \sum_{u} x_{p^*,u}(t) = \sum_{p^*} U_{p^*}, \quad p^* \in \mathcal{P}^*, \tag{1.d}$$

$$\sum_{p^{\otimes}} \sum_{u} x_{p^{\otimes},u}(t) = 0, \quad p^{\otimes} \in \mathcal{P}^{\otimes}. \tag{1.e}$$

In such an optimization problem, the aim is to maximize for each RB $b \in \mathcal{B}$ the sum of utility values over learners $u \in \mathcal{U}_p$ of class $p \in \mathcal{P}$ decided by the prioritization sequence in each TTI t. However, the wireless environment must be considered in the optimization problem to enable scheduling and resource allocation for users with high utility values and favorable channel conditions. Therefore, learner $u \in \mathcal{U}_p$ gets the RB $b \in \mathcal{B}$ if the metric

$$\Gamma_n\left[k_{p,u,n}(t)\right] \cdot \lambda_{u,b}(t)$$

is maximized relative to all other learners' metrics, where $\lambda_{u,b}(t)$ is the achievable rate that could be obtained if RB $b \in \mathcal{B}$ would be allocated to learner $u \in \mathcal{U}_p$ at TTI t.

To solve such complex problems, two variables must be determined each TTI t:

a) $x_{p,u} \in \{0, 1\}$ decides the learner $u \in \mathcal{U}_p$ to be scheduled in the frequency domain (i.e., if $x_{p,u} = 1$, then video class $p \in \mathcal{P}$ is prioritized and user $\forall u \in \mathcal{U}_p$ passed in the frequency domain; if $x_{p,u} = 0$, then video class $p \in \mathcal{P}$ is not prioritized);

b) $y_{u,b} \in \{0, 1\}$ performs the scheduling and resource allocation (i.e., if $y_{u,b} = 1$, then RB $b \in \mathcal{B}$ is allocated to learner $u \in \mathcal{U}_p$; if $y_{u,b} = 0$, then user $u \in \mathcal{U}_p$ does not receive $b \in \mathcal{B}$).

When obtaining the best combinations of users and RBs to maximize (1.a) each TTI, a set of constraints must also be considered. Therefore, constraints (1.b) indicate that each RB $b \in \mathcal{B}$ is allocated to one learner at most. Also, as requested by (1.c), once a video class $p \in \mathcal{P}$ is prioritized, all learners $u \in \mathcal{U}_p = \{u_1, u_2, \ldots, u_{U_p}\}$ within that class are competing to get the available resources allocated, where U_p is the number of learners in class $p \in \mathcal{P}$. In case of remaining resources after scheduling the higher prioritized class, the optimization problem is repeated for the next video class from the prioritized sequence. However, due to unfavorable networking conditions, some video classes can remain unscheduled at certain TTIs. In this sense, let us define by $\mathcal{P}^*(t)$ the set of video classes scheduled at TTI t, while by $\mathcal{P}^{\otimes}(t)$ we define the set of video classes remained unscheduled, where $\mathcal{P}^* \cup \mathcal{P}^{\otimes} = \mathcal{P}$ and $\mathcal{P}^* \cap \mathcal{P}^{\otimes} = \{\varnothing\}$. Accordingly, the constraints (1.d) show that all learners in the scheduled classes $p^* \in \mathcal{P}^*$ are passed in the frequency domain and compete for radio resource allocation. Meanwhile, the other learners in $p^{\otimes} \in \mathcal{P}^{\otimes}$ are deprived of receiving video packets in that TTI t due to the fact that there are not enough radio resources left after scheduling learners in $p^* \in \mathcal{P}^*$, as indicated by the constraints (1.e).

B. Problem Solving

To find optimal solutions in (1.a) in each TTI t, the scheduler needs to identify the best type of utility function n to be employed, and at the level of each resource block $b \in \mathcal{B}$, the most appropriate learner $u \in \mathcal{U}_p$ and service class $p \in \mathcal{P}$. This decision-making should be done in such a way that the set of constraints (1.b)-(1.e) are met in each TTI and the number of KPIs $k_{p,u,n}$ that satisfy their associated requirements $q_{p,n}$ is maximized in the subsequent TTI $t + 1$. This approach raises two main problems:

a) the decision process becomes time-consuming, as each possible combination $n \times b \times u \times p$ must be tested, and the best one has to be selected to perform scheduling;

b) finding the optimal solution in each TTI is complex, as the performance (meeting the QoS requirements) of each possible decision in *a)* needs to be known in advance.

Therefore, we want to simplify the solution-search problem at each TTI by finding sub-optimal solutions of the original optimization problem in two stages:

a) the prioritization sequence of video classes;

b) the scheduling of pre-selected learners and resource allocation by respecting the prioritization of video classes decided in *a)*.

To solve the first sub-problem, this paper employs a PriMARL-based solution to increase the QoS provisioning by deciding at each TTI the best prioritization of video classes. However, the type of scheduling rule used in resource allocation has a major impact in QoS and QoE provisioning for the pre-selected users. In this paper, we train our PriMARL method by employing three different scheduling rules in the frequency domain [30]: PriMARL-BF, PriMARL-EXP, and PriMARL-OPLF with their main focus on a particular QoS performance indicator, namely throughput ($n = 1$), delay ($n = 2$), and packet loss ($n = 3$), respectively.

IV. Proposed PriMARL Solution

A controller is employed in Fig. 2 to interact with the scheduler entity and learn the best prioritization decision to be taken at each TTI t. In a real system, this controller is deployed at the MAC layer of the 5G gNodeB base station and is owned by the network operator. The interaction between controller and scheduler at the level of MAC layer is modeled according to: *state* representing the observable data received from the scheduler, *action* corresponding to the prioritization sequence, and *reward* that measures in the current state how good the prioritization decision taken in the previous state is. By experiencing a very large amount of interactions in terms of previous state - action - reward - current state, the controller learns from trials and errors to improve its decisions over time based on reinforcement learning [31]. The controller considers P number of agents trained to compute the prioritization decision of video classes at each TTI t. In particular, each agent $p \in \mathcal{P}$ learns to claim at each TTI the priority of class $p \in \mathcal{P}$ to be passed in the frequency domain. Then, the controller computes a joint action by ordering the priority values given by each particular agent. Since each agent learns based on its own state to compute a joint action together with other agents, the proposed approach works in a multi-agent reinforcement learning mode [8]. We argue that combining the decisions of multiple agents with various priorities is more efficient than using a single agent that decides the prioritization sequence once at each TTI.

A. States, Actions, and Rewards

An instantaneous state of agent $p \in \mathcal{P}$ observed at TTI t is given by the data sample $\mathbf{s}_p(t) \in \mathcal{S}_p$, where $\mathcal{S}_p$ is the state space of class $p \in \mathcal{P}$. This state is divided into two parts:

$$\mathbf{s}_p(t) = \big[\mathbf{c}_p(t), \mathbf{n}_p(t)\big],$$

where $\mathbf{c}_p(t)$ are some controllable elements that can be influenced by the prioritization decisions, while $\mathbf{n}_p(t)$ are some non-controllable elements such as the Channel Quality Indicator (CQI) that changes regardless of the applied decision. The controllable sample is represented by

$$\mathbf{c}_p(t) = \Big[\mathbf{k}_p, \underline{\mathbf{k}}_p, \mathbf{d}_p\Big] \in \mathcal{S}_p^c,$$

where

$$\mathbf{k}_p = \Big[\mathbf{k}_{p,u_1}, \mathbf{k}_{p,u_2}, \ldots, \mathbf{k}_{p,u_{U_p}}\Big]$$

is the KPI vector of all learners in class $p \in \mathcal{P}$, $\underline{\mathbf{k}}_p$ is a vector that computes the differences between each KPI $k_{p,u,n}$ from vector $\mathbf{k}_p$ and its associated QoS requirement $q_{p,n} \in \mathcal{Q}_p$, and

$$\mathbf{d}_p = \Big[d_{p,u_1}, d_{p,u_2}, \ldots, d_{p,u_{U_p}}\Big]$$

is the vector containing the amount of queued data for each learner at the level of MAC layer. At each TTI t, the controller state $\mathbf{s}(t) \in \mathcal{S}$ is obtained by encompassing all agents' states:

$$\mathbf{s}(t) = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_P] \in \mathcal{S},$$

where $\mathcal{S}$ is the controller state space.

A joint action is denoted by

$$\mathbf{a}(t) = [a_i]_{i=1,2,\ldots,P} \in \mathcal{A},$$

where $a_i \in \mathcal{P}$ is the video class with the i^{th} priority to be scheduled at TTI t, and $\mathcal{A}$ is the P dimensional and discrete controller action space. As mentioned, a number of P^* classes can be used for scheduling, and consequently, the action

$$\mathbf{a}(t) = [a_1, \ldots, a_{P^*}, \ldots, a_P]$$

is partially used, where $1 \leq P^* \leq P$.

The controllable state of each agent evolves to the next states based on applied joint action:

$$\mathbf{c}'_{a_i} = f_{a_i}\big(\mathbf{s}_{a_i}, \mathbf{a}\big), \tag{2}$$

where

$$\mathbf{c}'_{a_i} = \mathbf{c}_{a_i}(t+1)$$

is a controllable state at TTI t+1, and

$$f_{a_i} : \mathcal{S}_{a_i}^c \times \mathcal{P} \rightarrow \mathcal{S}_{a_i}^c$$

is the transition function that moves the agent from the state $\mathbf{s}_{a_i}(t) \in \mathcal{S}_{a_i}$ to the next state $\mathbf{s}_{a_i}(t+1) \in \mathcal{S}_{a_i}$ when scheduling learners in class $\forall a_i \in \mathcal{P}$ at TTI t.

The reward function of the controller depicted in Fig. 2 measures the impact of applying action $\mathbf{a}(t) \in \mathcal{A}$ in state $\mathbf{s}(t) \in \mathcal{S}$, defined as [32]:

$$R(\mathbf{s}, \mathbf{a}) \overset{(\mathbf{def})}{=} \mathbb{E}\Big[R_{t+1} | \mathbf{s}(t) = \mathbf{s}, \mathbf{a}(t) = \mathbf{a}\Big], \tag{3}$$

where $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, and $\mathbb{E}[\cdot]$ is the expectation operator, with a random state $\mathbf{s}(t) \in \mathcal{S}$ so that, $\mathbb{P}[\mathbf{s}(t) = \mathbf{s}] > 0$ and $\mathbb{P}[\mathbf{a}(t) = \mathbf{a}] > 0$ hold for all $\mathbf{a} \in \mathcal{A}$. For our purpose, the reward function is computed as follows [7]:

$$R(\mathbf{s}, \mathbf{a}) = \sum_{i=1}^{P} \chi(a_i) \cdot r_{a_i}\big(\mathbf{s}_{a_i}, \mathbf{a}\big), \tag{4}$$

where

$$r_{a_i} : \mathcal{S}_{a_i} \times \mathcal{P} \rightarrow \mathbb{R}$$

is the reward function that evaluates the QoS performance when scheduling learners in video class $a_i \in \mathcal{P}$, and

$$\chi : \mathcal{P} \rightarrow [0, 1], \quad \chi(a_i) = (P + 1 - a_i) \Big/ \sum_{h=1}^{P} h$$

is the weight function that sets the importance of each reward r_{a_i} given the sequence $[1, 2, \ldots, P]$ requested by the prioritization standard. By using (2) and measuring the QoS performance in each video class, the proposed reward becomes:

$$r_{a_i}(\mathbf{s}_{a_i}, \mathbf{a}) \overset{(2)}{=} r_p\left(\mathbf{c}'_p\right) = \frac{1}{U_p} \cdot \frac{1}{N} \sum_u \sum_n r_{u,n}\left(\mathbf{c}'_{p,u,n}\right), \quad (5)$$

where we assume that action $a_i = p \in \mathcal{P}$, and $r_{u,n}$ is the particular reward of user $u \in \mathcal{U}_p$ and QoS requirement $q_n \in \mathcal{Q}$, with the function argument given by the controlable sample

$$\mathbf{c}'_{p,u,n} = \left[k'_{p,u,n}, \underline{k}'_{p,u,n}, d'_n\right].$$

As shown in [7], the computation of the learners' rewards $r_{u,n}$ depends on type n of QoS requirement for each traffic class.

B. Policy and Value Functions

The proposed solution considers the stochastic game with the tuple

$$\langle \mathcal{S}_1, \ldots, \mathcal{S}_P, \mathcal{A}, f_1, f_2, \ldots, f_P, R \rangle,$$

meaning that each agent $p \in \mathcal{P}$ learns based on its own state space $\mathcal{S}_p$ to cooperate with other agents to maximize the overall QoS provisioning in all video classes according to the employed reward functions in (4) and (5).

Each agent keeps its own policy function

$$\pi_p : \mathcal{S}_p \times \mathcal{A} \to [0, 1]$$

defined as the probability of selecting a given joint action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{s}_p \in \mathcal{S}_p$ [32]. Similar to the joint action, we compute the controller joint policy as the sequence of

$$\pi = \left[\pi_p\right]_{p=1,2,\ldots,P}.$$

Furthermore, each agent keeps track of an action-value function to calculate the expected cumulative future reward if agent $p \in \mathcal{P}$ is in state $\mathbf{s}_p$, executes the joint action $\mathbf{a} \in \mathcal{A}$ by obtaining the ith priority to be scheduled, and the joint policy π is subsequently followed. We define this function by [32]:

$$Q_p : \mathcal{S}_p \times \mathcal{A} \to \mathbb{R},$$
$$Q_p(\mathbf{s}_p, \mathbf{a}) = \mathbb{E}\Big[\sum_{t=0}^{T \to \infty} \gamma^t R_{t+1} | \mathbf{s}_p(0) = \mathbf{s}_p, \mathbf{a}(0) = \mathbf{a}, \pi\Big], \quad (6)$$

where $0 \leq \gamma \leq 1$ is a discount factor that gives more importance to the immediate rewards than to the later ones, and $\mathbb{E}[\cdot]$ is the expectation operator with the same properties as shown in (3). The action-value function of each agent $p \in \mathcal{P}$ is trained separately to claim the priority of the corresponding video class to be scheduled in the frequency domain. When the controller is trained and the action-value functions are considered optimal or near-optimal, an action $\mathbf{a} \in \mathcal{A}$ is selected with a sequence of probabilities of $\pi(\mathbf{a}) = [1, 1, \ldots, 1]$:

$$\mathbf{a} = \mathbf{solve}_{p \in \mathcal{P}}\left[Q^*_p(\mathbf{s}_p, \cdot)\right]_{p=1,2,\ldots,P}, \quad (7)$$

where Q^*_p is the trained function, and **solve** gives the descending order of all action values and returns the agents' indices.

In addition to the action-value functions of the individual agents $p \in \mathcal{P}$, we use the value function $V(\mathbf{s})$ that considers the initial controller state $\mathbf{s}(0) = \mathbf{s} \in \mathcal{S}$ and underlies the joint policy π afterwards [32]:

$$V : \mathcal{S} \to \mathbb{R},$$
$$V(\mathbf{s}) = \mathbb{E}\Big[\sum_{t=0}^{T \to \infty} \gamma^t R_{t+1} | \mathbf{s}(0) = \mathbf{s}, \pi\Big]. \quad (8)$$

The role of $V(\mathbf{s})$ is to coordinate agents in the training process to learn the best prioritization decisions. In addition, the transition between two consecutive states can also be used based on [31]:

$$V(\mathbf{s}) = R(\mathbf{s}, \mathbf{a}) + \gamma \cdot V(\mathbf{s}'), \quad (9)$$

where $\mathbf{s}' = \mathbf{s}(t+1) \in \mathcal{S}$ represents the next state. With these consecutive states $\{\mathbf{s}, \mathbf{s}'\} \in \mathcal{S}$ and reward function $R(\mathbf{s}, \mathbf{a})$, the value of the previous state $V(\mathbf{s})$ is updated based on (9).

C. Solution Employment

In order to use the proposed solution in real-time systems, two major aspects need to be considered:

a) the dimension of all states $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_P\}$ depends on the number of active learners $\{U_1, U_2, \ldots, U_P\}$ that can change over time;

b) because of the multi-dimensionality of the state space, the action-value and value functions cannot be updated using conventional look-up tables.

Therefore, we address these challenges through compression and approximation methods, respectively.

The original state space $\mathcal{S}_p$ is compressed to avoid the dependency on U_p by applying the transformation:

$$\bar{\mathcal{S}}_p = \mathcal{T}(\mathcal{S}_p), \quad (10)$$

where $\mathcal{T}$ is the transformation operator and $\bar{\mathcal{S}}_p$ is the compressed state space of class $p \in \mathcal{P}$ of constant dimension over a variable number of mobile learners. Depending on the elements in $\mathbf{s}_p \in \mathcal{S}_p$, the space transformation can have different computations. For example, descriptive statistics (mean and standard deviation) are used for the vector of controllable elements

$$\left[k_{p,u,n}, \underline{k}_{p,u,n}, d_u\right]$$

for all $p \in \mathcal{P}$, $u \in \mathcal{U}_p$ and $n \in \{1, 2, \ldots, N\}$ [29]. In case of non-controllable elements (e.g., CQI), unsupervised and supervised learning techniques are used [29].

With the compression mechanism, the obtained states $\bar{\mathbf{s}}_p \in \bar{\mathcal{S}}_p$ are still multi-dimensional and function approximators must be used to model the action-value and value functions. In this paper, we adopt the use of feed-forward neural networks as parameterizable functions to be learned over time to provide the best prioritization sequence on each state. Therefore, each agent $p \in \mathcal{P}$ is represented by

$$Q_p(\bar{\mathbf{s}}_p, \mathbf{a}; \Theta_p) \approx Q_p(\bar{\mathbf{s}}_p, \mathbf{a}),$$

where Θ_p is the set of weights that must be updated during the training stage. To increase the training efficiency of the

proposed solution, we also employ the value function of the controller state $\bar{\mathbf{s}} \in \bar{\mathcal{S}}$ and approximated by the neural network

$$V(\bar{\mathbf{s}}; \Theta) \approx V(\bar{\mathbf{s}}).$$

Therefore, a number of $P+1$ neural networks must be trained during the learning stage of the proposed PriMARL solution.

During training, a joint action $\mathbf{a} \in \mathcal{A}$ is selected by each agent on each state $\bar{\mathbf{s}}_p \in \bar{\mathcal{S}}_p$ according to:

$$\pi_p(\mathbf{a} \mid \bar{\mathbf{s}}_p) = \begin{cases} 1-\epsilon & \mathbf{a} = \mathbf{solve}\big[Q_p(\cdot; \Theta_p)\big]_{p=1,\ldots,P}, \\ \epsilon & \mathbf{a} = \mathbf{solve}\big[\mathbf{rand}_p\big]_{p=1,\ldots,P}, \end{cases} \quad (11)$$

where $\mathbf{rand}_p \in [0, 1]$ is a sequence of random numbers. In some cases, parameter $\epsilon \in [0, 1]$ is set to higher values at the beginning of the training stage (more exploration in terms of the random action selections), and to lower values at the end of the training (more exploitation based on the trained functions). In some other cases, ϵ can have constant value for the entire training period. Regardless of the strategy used, the same value of ϵ is used by all agents at each TTI.

Once a joint action $\mathbf{a} \in \mathcal{A}$ is applied at TTI t, the system moves to the next state, and a reward $R(\bar{\mathbf{s}}, \mathbf{a})$ is computed. We denote by

$$E(t+1) = \big\{\bar{\mathbf{s}}, \mathbf{a}, R, \bar{\mathbf{s}}', P^*(t)\big\}$$

the controller experience at TTI $t+1$, and $P^*(t)$ is the number of classes scheduled at TTI t. The experience of an agent $p \in \mathcal{P}$ is given by

$$E_p(t+1) = \big\{\bar{\mathbf{s}}_p, \mathbf{a}, \bar{\mathbf{s}}'_p\big\}.$$

All these experiences $e \in \{E, E_1, E_2, \ldots, ., E_P\}$ are used at each TTI to reinforce the neural networks with the aim of minimizing the following cost function:

$$\mathcal{C}(\theta) = \mathbb{E}_{e(t)}\left\{\frac{1}{2}[\eta \cdot \delta(\theta)]^2\right\}, \quad (12)$$

where $\eta \in [0, 1]$ is the learning rate,

$$\theta \in \{\Theta, \Theta_1, \Theta_2, \ldots, \Theta_P\}$$

is the set of weights of the trained neural networks, and $\delta(\theta)$ is the Temporal Difference (TD) error computed as a difference between the target and the actual estimate of the network:

$$\delta(\theta) = F^T(\cdot; \theta) - F(\cdot; \theta). \quad (13)$$

By $F(\cdot; \theta)$ we mean both the functions $V(\cdot; \Theta)$ and $Q_p(\cdot; \Theta_p)$ for all $p \in \mathcal{P}$. The target $F^T(\cdot; \theta)$ is determined separately for value and action-value functions. For example, the target of value function takes the form of (9) and the TD error becomes

$$\delta(\Theta) = V^T(\bar{\mathbf{s}}; \Theta) - V(\bar{\mathbf{s}}; \Theta).$$

We design the neural network that learns the value function as a critic to determine whether the multi-agent system decision is a good or bad option. If $\delta(\Theta) \geq 0$, the prioritization sequence $\mathbf{a} \in \mathcal{A}$ has a positive effect and the cost values should be reinforced in the networks with a relatively higher learning rate $\eta = \alpha$. If $\delta(\Theta) < 0$, such actions must be prevented in the future by using a lower learning rate $\eta = \beta$ and thus, $\beta << \alpha$, when choosing the parameters of the PriMARL controller.

Even when the TD error becomes positive, the prioritization decision can infuse the over-provisioning effect and some classes with met QoS requirements ($r_p = 1$) are prioritized at the expense of other classes with unmet QoS requirements ($r_{p'} < 1$), $\forall p \neq p' \in \mathcal{P}$. To address this problem, we employ

$$h : \mathcal{P}^* \times [0, 1]^P \rightarrow \{0, 1\}$$

as a penalty function to improve the decision-making, so that:

a) if $h(a_{i^*}, r_1, \ldots, r_P) = 1, i^* = 1, 2, \ldots, P^*$, then all video classes $a_{i^*} \in \mathcal{P}^*$ meet the QoS requirements but are prioritised at the expense of other classes whose QoS requirements are not met and whose rewards are lower than $r_{p'} < 1$;

b) if $h(a_{i^*}, r_1, \ldots, r_P) = 0, i^* = 1, 2, \ldots, P^*$, then prioritising $a_{i^*} \in \mathcal{P}^*$ among other classes is a fair choice. Then, the proposed target of the action-value function becomes:

$$Q^T_{a_{i^*}} = \begin{cases} \frac{P}{(P+1-i^*)}\ ,\ \eta = \alpha & \text{if } \delta \geq 0 \text{ and } h(\cdot) = 0, \\ -0.5\ ,\ \eta = \alpha & \text{if } \delta \geq 0 \text{ and } h(\cdot) = 1, \\ -1\ ,\ \eta = \beta & \text{if } \delta < 0, \end{cases} \quad (14)$$

where $Q^T_{a_{i^*}}(\bar{\mathbf{s}}_{a_{i^*}}, \mathbf{a}; \Theta_p)$ is the target function of those classes $a_{i^*} = p^* \in \mathcal{P}^*$ being scheduled at TTI t, while the rest of the agents are not updated. As observed in (14), negative target values are associated even when the value function error is positive ($\delta(\Theta) \geq 0$), but the penalty function shows inequity between prioritized video classes ($h(\cdot) = 1$). Therefore, the error to be reinforced by the agent $p^* \in \mathcal{P}^*$ averaged with the learning rate $\eta = \{\alpha, \beta\}$ becomes

$$\delta_{p^*}(\Theta_{p^*}) = Q^T_{p^*}(\cdot; \Theta_{p^*}) - Q_{p^*}(\cdot; \Theta_{p^*}).$$

Finally, the weights of the critic neural network and all agents are updated based on the Stochastic Gradient Descent (SGD) algorithm, which is given by the following formula [7]:

$$\theta \leftarrow \theta + \eta \frac{\partial F}{\partial \theta}(\cdot; \theta) \cdot \delta(\theta). \quad (15)$$

In Algorithm 1, we describe how PriMARL is trained to prioritize traffic classes and allocate radio resources through a specific scheduling rule based on utility function Γ_n. As input parameters, the algorithm considers two consecutive states $\{\mathbf{s}, \mathbf{s}'\} \in \mathcal{S}$, the action applied in the previous state $\mathbf{a} \in \mathcal{A}$, and the number of video classes $P^*(t)$ being scheduled in the previous state. As an output, Algorithm 1 provides a new action $\mathbf{a}' \in \mathcal{A}$ as a prioritization sequence and executes the scheduling and allocation of radio resources. In the first step (lines (4)-(8)), the controller's reward is calculated, the states are compressed for each agent, the error of the value function (critic) is back-propagated, and the weights are updated based on the SGD algorithm. We set different learning rates for the agents if the critic error is positive or negative (line 10). In the second step, we update the agents representing the traffic classes that were scheduled in the previous TTI (lines (11)-(15)). In the third step, the video classes are prioritized according to the new joint action $\mathbf{a}' \in \mathcal{A}$ decided by all agents (line 17). In the frequency domain, radio resources in $\mathcal{B}$ are allocated to prioritized learners competing with each other based on the type of utility function Γ_n or scheduling

Algorithm 1 PriMARL Training in Traffic Prioritization and Scheduling With a Particular Utility Function Γ_n

1: **input:** $\mathbf{s} \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}$, $\mathbf{s}' \in \mathcal{S}$, $P^*(t)$
2: **output:** $\mathbf{a}' \in \mathcal{A}$, scheduling and radio resource allocation
3: **for** each TTI t+1
4: **calculate** rewards based on (3)-(5)
5: **compress** states $\{\mathbf{s}_p, \mathbf{s}'_p\}_{p=1,...,P}$ and $\{\mathbf{s}, \mathbf{s}'\}$
6: **recall** experiences $\{E_1, E_2, ..., E_P, E\}$
7: **calculate** the value function error $\delta(\Theta)$ based on (13)
8: **back-propagate** $\delta(\Theta)$ and update weights based on (15)
9: // **criticize** previous action $\mathbf{a} \in \mathcal{A}$
10: **if** $\delta(\Theta) \geq 0$, then $\eta = \alpha$, else $\eta = \beta$
11: **for** $i^* = 1, 2, \ldots, P^*$
12: **determine** target function $Q^T_{a_{i^*}}$ based on (14)
13: **calculate** error $\delta_{a_{i^*}}(\Theta_{a_{i^*}})$ based on (13)
14: **back-propagate and update** $\Theta_{a_{i^*}}$ based on (15)
15: **end for**
16: // **act** based on the joint policy
17: **determine** new action $\mathbf{a}' \in \mathcal{A}$ based on policy (11)
18: **while** $\mathcal{B} \neq \varnothing$
19: **pick** video class $a'_i = p$, $\forall p \in \mathcal{P}$
20: **perform** scheduling based on (1.a)-(1.e)
21: **add** $a'_i = p$ in the set of scheduled video classes $\mathcal{P}^*$
22: $i = i + 1$
23: $P^* = P^* + 1$
24: **end while**
25: **end for**

Algorithm 2 PriMARL Testing in Traffic Prioritization and Scheduling With a Particular Utility Function Γ_n

1: **input:** states $\mathbf{s} \in \mathcal{S}$
2: **output:** $\mathbf{a} \in \mathcal{A}$, scheduling and radio resource allocation
3: **for** each TTI t
4: **compress** states $\{\mathbf{s}_p\}_{p=1,...,P}$ and $\mathbf{s}$
5: **for** $p = 1, 2, \ldots, P$
6: **determine** output $Q_p(\cdot; \Theta_p)$ of agent $p \in \mathcal{P}$
7: **end for**
8: // **prioritize** based on the joint action
9: **determine** new action $\mathbf{a} = \textbf{solve}[Q_p(\cdot; \Theta_p)]_{p=1,...,P}$
10: **while** $\mathcal{B} \neq \varnothing$
11: **pick** video class $a_i = p$, $\forall p \in \mathcal{P}$
12: **perform** scheduling based on (1.a)-(1.e)
13: $i = i + 1$
14: **end while**
15: **end for**

rule used (i.e., BF, OPLF, EXP). Learners from the prioritized list

$$\mathbf{a}'(t) = \left[a'_i\right]_{i=1,2,...,P} \in \mathcal{A}$$

have access to radio resources within the limits of the available stock (lines (18)-(24)). For example, learners in class $a'_2 \in \mathcal{P}$ compete for radio resources if there are enough resources left after scheduling the higher-priority class $a'_1 \in \mathcal{P}$ in the sequence.

In Algorithm 2, each PriMARL algorithm is tested and implemented in real-time scheduling. Here, the process is simplified, since only the current states $\mathbf{s} \in \mathcal{S}$ are needed as input parameters, and the algorithm will provide a new prioritization sequence given by the trained agents. The neural networks are no longer updated, but the algorithm still needs to compress the states (line 4) of each agent $p \in \mathcal{P}$. The joint action is decided at each TTI by ordering the agents' outputs (lines (5)-(9)), and the scheduling process is performed based on (1.a)-(1.e), depending on the type of scheduling rule Γ_n and the available stock of radio resources (lines (11)-(14)).

V. Simulation Results

The proposed PriMARL framework is developed in a C/C++ software environment using intelligent OFDMA scheduling in both the time and frequency domains, data compression mechanisms, and neural networks to approximate agents' decisions for each video class. The proposed tool inherits the LTE-Sim functionality [33]. As explained above, the proposed PriMARL-based solution considers three types of utility functions as scheduling rules [6]: PriMARL-BF, PriMARL-OPLF, and PriMARL-EXP. Since most of the state-of-the-art works presented in Section II do not provide the level of detail necessary to enable their implementation, we provide a comprehensive comparison of the proposed solutions with the following approaches: HiMARL [7], FLS [24], and RADS [25]. We evaluate the performance of these schedulers from the perspective of:

a) QoS provisioning, where throughput, delay, and packet loss indicators are monitored in each TTI. To quantify the level of QoS provisioning in the time domain, three types of QoS requirements are considered for each video class: Guaranteed Bit Rate (GBR, $n = 1$), packet delay ($n = 2$), and Packet Loss Rate (PLR, $n = 3$).

b) QoE provisioning by calculating the perceived PSNR based on throughput and arrival rates. As a result of PSNR assessment, MOS is calculated on five different levels: *excellent (5)*, *good (4)*, *fair (3)*, *poor (2)*, and *bad (1)*.

The purpose of this section is to demonstrate that setting a multi-objective target ($n = \{1, 2, 3\}$) to maximize the QoS provisioning does not guarantee the same effect in terms of perceived PSNR and MOS levels. In particular, we show that the proposed PriMARL solution is able to outperform HiMARL, RADS, and FLS when monitoring the number of learners achieving excellent MOS levels while viewing different types of educational video content. Therefore, we organise this section as follows: *a)* first, we present the traffic characteristics, network, scheduler and controller settings; *b)* then, we present the QoS analysis in terms of throughput, delay, packet loss, and the number of TTIs when all three QoS objectives are met. *c)* In the third part, QoE analysis is performed for PSNR and MOS levels for all approaches which are involved in this comparison framework. *d)* Finally, we provide additional results and insights to better highlight the importance of using PriMARL approaches with static scheduling rules from the perspective of QoE performance.

A. Video Traffic Settings

As shown in Fig. 1, learners access the heterogeneous video contents from mobile devices. To cope with the different financial situations of learners, in this study we consider two resolutions of mobile devices, 240p and 480p, linked to lower and higher prices, respectively. According to [34], for each resolution, maximum thresholds for low and high bit rate values are recommended: *a)* for 240p, 150kbps and 250kbps; while

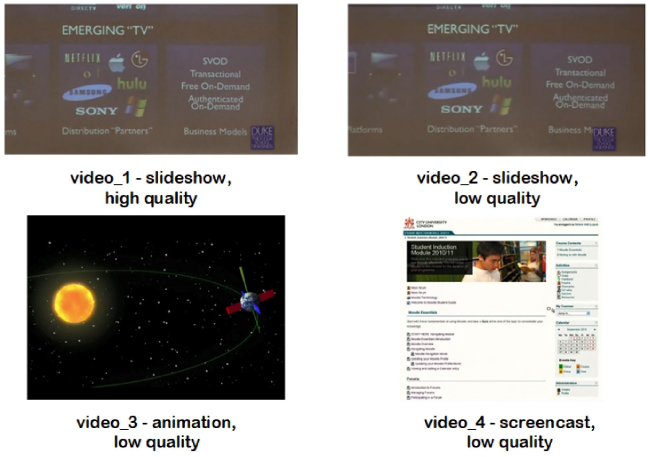

Fig. 3. Example of Educational Content Video Classes.

for 480p, maximum rates of 0.6Mbps and 1Mbps are recommended. Based on the subjective surveys conducted in [7], learners were asked to rate video quality using mean opinion scores for seven categories of educational videos with low and high quality levels. All content categories with low quality levels were perceived as good by all viewers, with the exception of slideshow content, which was perceived as fair. Similarly to [7], we consider the same classes of video services, i.e., low and high quality slideshows with a resolution of 240p, as well as animations and screencasts for devices with a resolution higher than 480p. By modeling animation as video traffic with a variable bit rate and screencast video with a constant bit rate, as well as by standardizing the QoS requirements [28], we obtain $P = 4$ video classes with the following characteristics:

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

Figure 3 illustrates an example of a video frame from each educational video class considered.

In such environments, the role of PriMARL is to increase the QoS provisioning in all classes by learning the best prioritization sequence to apply at each TTI according to the actual traffic and networking conditions. We then study the impact of this dynamic prioritization and different scheduling rules (BF, OPLF, EXP) on the QoE metrics, namely PSNR and MOS. During the training and testing stages, the aggregate traffic load of all classes is varied in an interval of $u \in [6.60]$, while respecting the following ratios between video classes: `video_1` (16.53%), `video_2` (16.53%), `video_3` (33.3%), and `video_4` (33.3%). Then, the QoS and QoE performance is evaluated based on three traffic load settings: low ($U \in [6, 20]$), medium ($U \in [21, 40]$), and high ($U \in [41.60]$). All scheduling approaches are tested for each configuration of U, and then, the results are averaged over the number of possible configurations in each traffic setting.

B. Network Settings

From the network perspective, we consider downlink scheduling sessions over the OFDMA interface with a system bandwidth of 20MHz and a number of $B = 100$ RBs. The radio channel model uses fast fading based on Jakes' model due to the high diversity provided in the CQI reports necessary to employ unsupervised learning techniques to find patterns and supervised learning methods to automate the CQI compression process [29]. The most widely used 7-cell cluster inter-cell interference model is considered. Each cell follows a macro-urban model with a radius of 1 km, since a wide range of CQI reports should be captured. When training the PriMARL controller, we consider a generic speed of 30km/h to teach the neural networks how to behave under different channel conditions, while when testing its performance, we consider static positions of the learners over several trials, as explained in more detail later in this section. We neglect intra-cell interference between mobile devices and other electronic devices, as this aspect is not relevant to our study. When training the machines, all ML-based approaches (HiMARL, PriMARL-BF, PriMARL-OPLF, PriMARL-EXP) are trained separately with different networking conditions. In the test phase, all candidates use the same network conditions.

C. Packet Scheduler Settings

At the level of the packet scheduler, the modulation and coding scheme is adapted at three levels (QPSK, 16-QAM, and 64-QAM) and the scheduling is done at each TTI in the time and frequency domain. In the radio link protocol layer, video packets are transmitted in acknowledged mode, with a maximum of five re-transmissions allowed for each lost packet. Once the scheduling process is complete and the system moves to the next TTI, the QoS indicators obtained are compared with the QoS requirements for each video traffic to verify the level of QoS provisioning. The delay of learner u requesting one of the video services is measured as the head-of-line packet delay and should not be greater than the requirement. The packet loss and the throughput performance are measured by averaging all instantaneous lost packets and throughput, respectively, in a sliding time window of 1000 TTIs. Depending on the method used, scheduling in the time and frequency domains is performed based on different metrics:

a) Time-domain scheduling: On one hand, the PriMARL and HiMARL approaches prioritize learners from the same class by deciding the sequence of classes to schedule at each TTI. On the other hand, FLS and RADS prioritize learners from different video classes based on different metrics. For example, as explained in Section II, in time-domain scheduling, the FLS scheduler estimates the amount of real-time data to be transmitted in the next frame of 10 TTIs based on discrete linear control theory arguments. Then, the learners from different classes are prioritized based on the approximated quota of data needed to meet the delay requirements. In the case of RADS, learners requesting different video services are ranked based on a metric that considers fairness, delay, and throughput.

b) Frequency-domain scheduling: The proposed PriMARL approach uses different scheduling strategies to allocate data

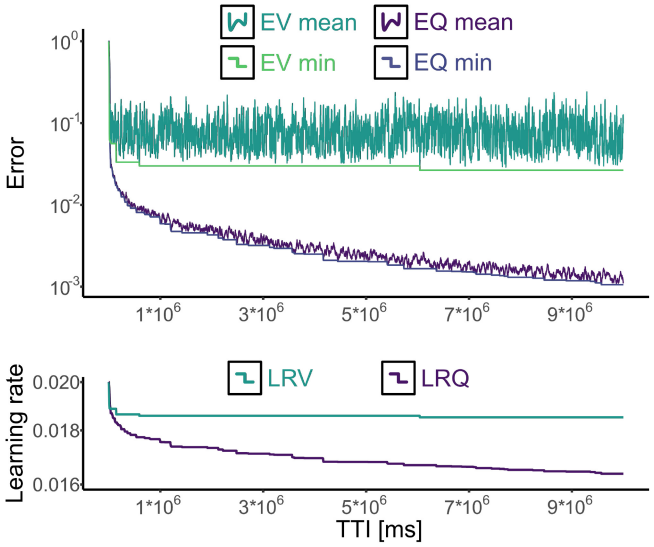

Fig. 4. PriMARL Training: Errors and Parameter Settings.

in the frequency domain, namely BF, OPLF, and EXP rules. HiMARL uses reinforcement learning solutions at the level of each video class to learn the best rule to apply each time that class is selected in the prioritization sequence. As the results will show, this scheme is able to balance the QoS provisioning between the PriMARL with separate scheduling rules, by affecting the QoE performance in terms of perceived video quality. In case of FLS, the proportional-fair scheduler is used in the frequency domain to improve the fairness between the pre-selected learners, while RADS uses OPLF to improve the PLR performance since this QoS indicator is not part of the metric used in the time domain.

The scheduling performance is assessed by comparing the six candidates based on different metrics. As performances can vary depending on the network and channel conditions, different trials are conducted, with all schedulers using the same conditions (number of learners, mobility, channel and traffic characteristics) in each trial to allow a fair comparison. Subsequently, the performance metrics are averaged using the following formula:

$$\mu_p(\mathbf{m}_p) = 1/G \cdot \sum_{g=1}^{G} \mathbf{m}_{p,g}, \tag{16}$$

where G is the number of trials in the test stage and $\mathbf{m}_{p,g}$ is the metric that evaluates the performance of a given indicator (QoS or QoE) for each video class $p \in \mathcal{P}$ in one trial g. In this study, we consider a number of $G = 10$ trials, where each trial has a duration of the scheduling process of about 50s.

D. PriMARL Controller Settings

The PriMARL controller is trained for a duration of 10^7 TTIs and the number of learners switched randomly from IDLE to ACTIVE and vice-versa every 1000 TTIs, taking into account the traffic load ratio between classes. To improve the generalization in decision-making, the speed of each mobile learner is set to 30kmph. Several configurations of neural networks were tested, and only the best ones are considered in this paper. For example, each neural network used to approximate an agent's ranking decision uses a hidden layer with 80 hidden nodes. When covering the entire state of all video classes, the value function uses a neural network with one hidden layer and 200 hidden nodes. In our settings, we choose a discount factor of $\gamma = 0.99$, which gives more importance to the value of the next-state when calculating the target value based on (9). Throughout training, we also consider equal chances of selection between exploration (random actions) and exploitation (actions based on trained functions) by setting $\epsilon = 0.5$. The learning rates for the critic and all agents are varied according to the minimum errors found during the training period.

Figure 4 shows the convergence analysis of the PriMARL algorithm in terms of mean and minimum errors and learning rates. By EV mean we denote the TD error of the critic neural network $\delta(\Theta)$ averaged over 1000 TTIs, and by EV min the minimum value reached during training. By EQ mean, we denote the error averaged over all agents and 1000 TTIs $(1/4000 \sum_{p=1}^{P} \delta(\Theta_p))$, while EQ min denotes the minimum value. It is worth noting that each time a new minimum is found in the mean error of each agent $p \in \mathcal{P}$, the set of weights Θ_p is stored. When evaluating the PriMARL approaches, the most recently stored set of weights is used. As can be seen in Fig. 4, the error of critic neural network drops below the value of 0.1 and remains relatively constant for the rest of the training period. In contrast, the mean error of all four agents converges to a value of 10^{-3} by the end of the training period. The learning rates associated with the critic (LRV) and agent (LRQ) neural networks are set to an initial value of 0.02 at the beginning of the training period, and gradually decrease with a step of 10^{-7} each time a new minimum error is found for each type of neural network.

E. QoS Analysis

To analyse the performance of QoS indicators in all video classes, we measure the levels of throughput, delay, and PLR for low, medium, and high traffic loads when employing the proposed PriMARL and state-of-the-art scheduling solutions. When quantifying the QoS provisioning, we are particularly interested in counting the number of TTIs when all QoS requirements are met in each video class.

1) Throughput, Delay, and Packet Loss: are collected for each scheduling scheme, traffic class, and mobile learner during the entire period of each trial. In particular, we are interested in calculating the percentiles for each collection of QoS indicators and identifying the worst indicators that could help us distinguish between the PriMARL solutions and other scheduling techniques. In this sense, we measure the percentiles of 5^{th} throughput, 95^{th} delay, and 95^{th} packet loss in each video class and average them over $G = 10$ trials.

Figure 5 (first row) shows the performance of scheduling candidates when monitoring the 5^{th} throughput percentile. For `video_1` and `video_2`, similar throughput is achieved by all solutions at low traffic load. However, for `video_3` and `video_4`, HiMARL, PriMARL-BF, PriMARL-OPLF,

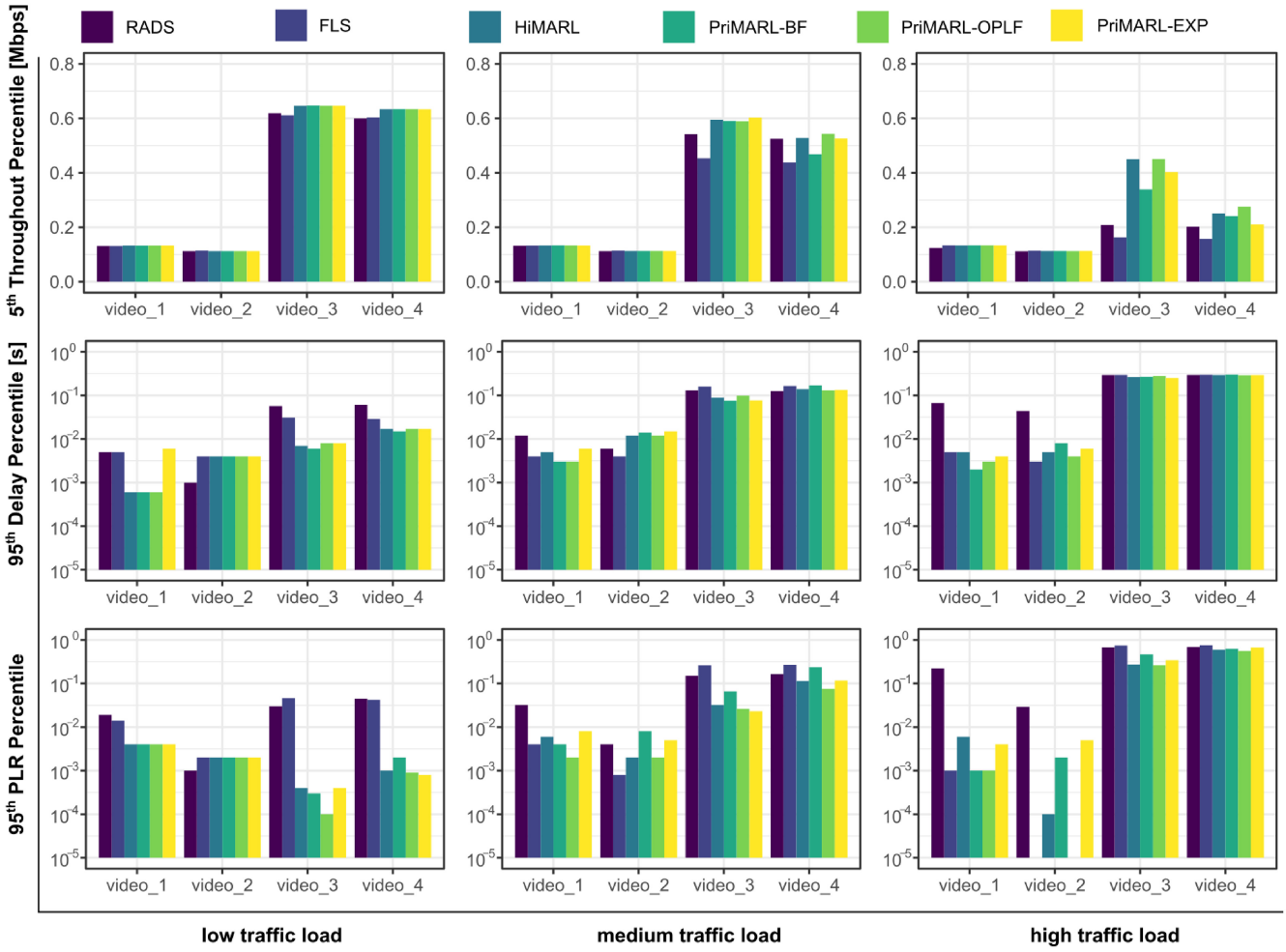

Fig. 5. The QoS performance evaluation averaged over $G = 10$ trials when measuring: the 5^{th} throughput percentile; the 95^{th} delay percentile; and the 95^{th} PLR percentile, each for low, medium, and high traffic load.

and PriMARL-EXP improve the level of the 5^{th} throughput percentile by about 30kbps compared to the non-ML candidates RADS and FLS. At medium traffic load, PriMARL-EXP is the best option in the `video_3` class, while in the `video_4` class PriMARL-OPLF outperforms PriMARL-EXP by more than 20kbps. By increasing the traffic load to 'high', in the first two prioritized video classes the throughput level remains nearly similar in both cases. A larger discrepancy in performance between ML and non-ML approaches could be observed in the case of `video_3`, where PriMARL-OPLF outperforms the FLS scheduler by more than 100kbps. The impact of dynamic prioritization of PriMARL schemes can be observed when comparing the throughput performance of the classes `video_3` and `video_4`. In this case, it can be observed that PriMARL-BF, PriMARL-OPLF, PriMARL-EXP, and HiMARL allocate a higher amount of resources to learners in the `video_3` class, while RADS and FLS are not able to prioritize `video_3` over `video_4`, achieving nearly the same throughput for both video classes. Except in the case of `video_3` with medium traffic load, the PriMARL-OPLF solution remains the best option when measuring the 5^{th} throughput percentile in all traffic classes.

Considering the high traffic load and summing up the 5^{th} throughput percentiles across all four traffic classes and for each scheduler, we obtain gains higher than 55% and 36% when comparing PriMARL-EXP with RADS and FLS, respectively. As we discussed in Section II, MDV [26] and NRADS [27] achieve throughput gains of about 25% and 10% when compared to FLS and RADS, respectively. Therefore, we can estimate the throughput gains of about 45% and 10% when comparing the PriMARL-EXP approach with the recent state-of-the-art schedulers NRADS and MCV, respectively.

The delay performance in terms of 95^{th} percentile is evaluated in Fig. 5 (middle row) for low, medium, and high traffic loads. In the first case, it can be observed that ML-based approaches perform better than RADS and FLS, especially for the `video_3` and `video_4` classes. Among all options, PriMARL-BF has the lowest delay in all video classes. When the traffic load is increased to medium and high, the delay increases, especially in `video_3` and `video_4`. At medium traffic load, PriMARL-BF and PriMARL-OPLF minimize the delay in `video_1`, FLS in `video_2`, PriMARL-EXP in `video_3` and RADS in `video_4`. For high traffic load, PriMARL-BF, FLS, PriMARL-EXP, and PriMARL-OPLF are the best solutions in the `video_1`, `video_2`, `video_3` and

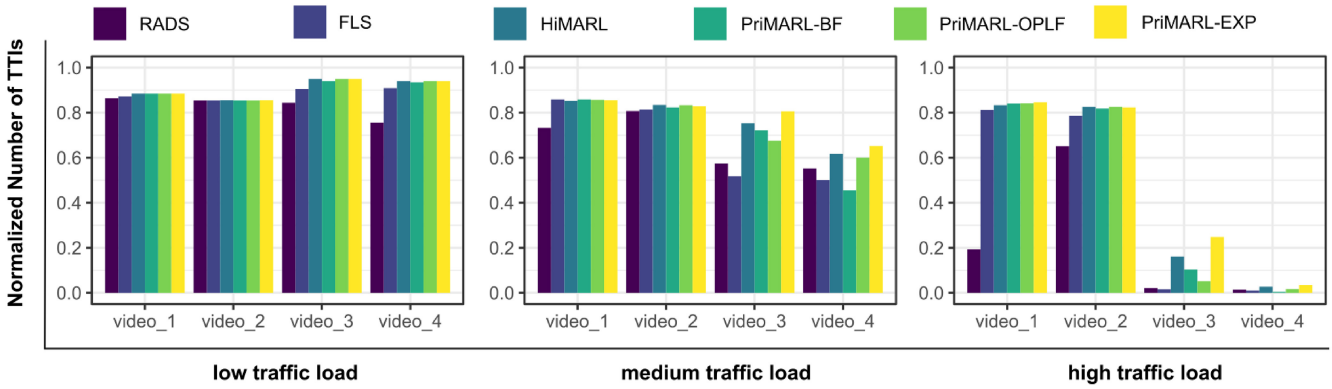

Fig. 6. The mean normalized number of TTIs when all QoS requirements (throughput, delay, PLR) are met in each video class for low, medium, and high traffic loads.

video_4 classes, respectively. However, when correlating the delay and throughput performance (Fig. 5 first and second rows), it is generally observed that lower delay percentiles are associated with higher throughput levels.

As shown in Fig. 5 (third row), the PLR performance is measured by the 95^{th} percentile of packet losses, averaged over the number of $G = 10$ trials. In case of low traffic load, the MARL approaches outperform RADS and FLS in video_1, while almost the same performance is obtained in video_2. In other traffic classes (video_3 and video_4), PriMARL-OPLF generally remains the best option among all candidates when scheduling low traffic load. For medium traffic load, PriMARL-OPLF gets the minimum PLR in video_1 and video_4, FLS in video_2, and PriMARL-EXP in video_3. When increasing the traffic load to high, the lowest PLR level is obtained by PriMARL-OPLF in all video classes. Similar to delay and throughput, when correlating the packet loss and user throughput, we observe that lower PLR involves higher throughput in terms of 5^{th} percentile and vice versa.

Looking at the performance of the QoS indicators in Fig. 5, we notice that PriMARL-OPLF generally performs better when measuring the 5^{th} throughput and the 95^{th} PLR percentiles, with a few exceptions. These exceptions relate to the PriMARL-EXP solution, which performs better in video_3 and video_4 when scheduling medium and low traffic loads, respectively. By using the reward as a multi-objective function of throughput, delay, and PLR, the HiMARL approach achieves a better balance of QoS performance in all video classes compared to the PriMARL approach with static rules. However, it remains to be verified whether this method is the best option for measuring duration when all QoS requirements are met in each video class.

2) Duration of QoS Provisioning: Figure 6 shows the normalized number of TTIs when all QoS requirements are met in each video class, averaged over $G = 10$ scheduling trials. In low traffic load settings, PriMARL and HiMARL approaches perform better compared to FLS and RADS schedulers. Since the video_1 and video_2 classes have a higher variability in arrival rates ($242kbps$ and $138kbps$, respectively) compared to animation and screencast videos (video_3 and video_4), it is very difficult to maintain certain levels of average throughput for these classes (video_1 and video_2) at the imposed GBR requirements over a very long period of time. This explains the longer duration of QoS provisioning in classes video_3 and video_4 compared to video_1 and video_2 when a low traffic load is scheduled. When the traffic load increases to medium and high, it is observed that the duration of QoS provisioning in video_1 and video_2 is similar to the previous case for all scheduling approaches, except for RADS where a higher performance degradation is obtained. However, in higher rate classes such as video_3 and video_4, PriMARL-EXP, PriMARL-OPLF, and HiMARL maintain the duration of providing high QoS significantly longer compared to FLS and RADS. In both settings of medium and high traffic loads, PriMARL-EXP is the best option, followed by HiMARL and PriMARL-BF in video_3 and PriMARL-OPLF in video_4. Therefore, the best strategy to maximise the duration of QoS provisioning is to schedule learners with the highest delay in each video class given the prioritization sequence decided for each TTI.

F. QoE Analysis

When we analyse the quality of experience of each learner being scheduled in each video class, we calculate the perceived PSNR at each TTI by employing the following formula [35]:

$$PSNR[dB] = 20 \cdot log_{10} \cdot \frac{R_{p,u}}{|R_{p,u} - T_{p,u}|}, \tag{17}$$

where $R_{p,u}$ is the arrival rate in the data queue and $T_{p,u}$ is the throughput of learner u receiving video services from class $p \in \mathcal{P}$. The PSNR levels are collected during each trial from each learner and at each TTI. In each trial, we compute the associated percentiles from the collected PSNR values. We then calculate the number of percentiles associated with each MOS level, starting with the worst PSNR percentile. Based on the calculated PSNR values, the MOS levels are determined as follows [36]: *Excellent* if $PSNR[dB] \geq 36$; *Good* if $29 \leq PSNR[dB] < 36$; *Fair* if $24 \leq PSNR[dB] < 29$; *Poor* if $20 \leq PSNR[dB] < 24$; and *Bad* if $PSNR[dB] < 20$. We average the percentage of MOS levels over $G = 10$ trials and present the results for low, medium, and high traffic loads.

1) Perceived PSNR: Since we could not differentiate between the MARL-based scheduling candidates in the

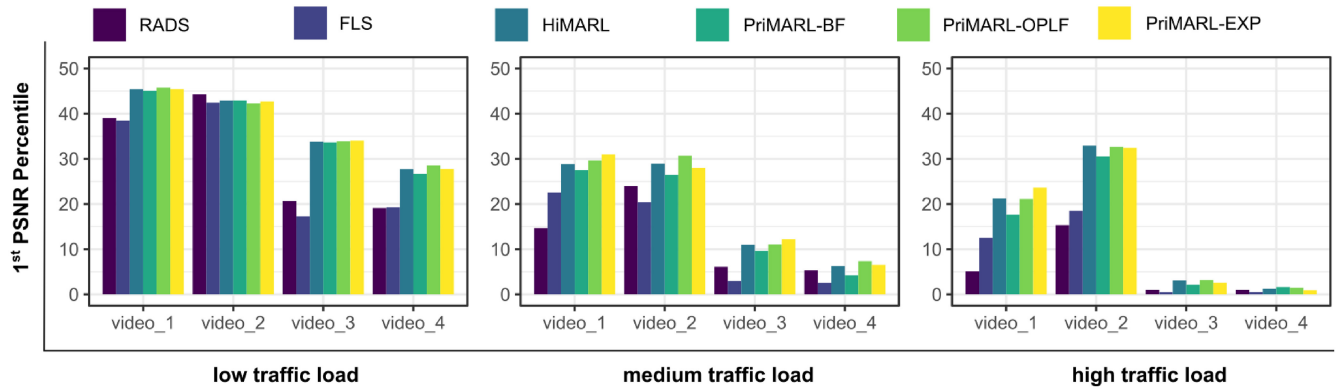

Fig. 7. Worst PSNR (1^{st} percentiles) levels averaged over the ten data sets for low, medium, and high traffic load.

TABLE I
MOS LEVELS FOR LOW TRAFFIC

Traffic	MOS	RADS (SD)	FLS (SD)	HiMARL (SD)	PriMARL -BF (SD)	PriMARL -OPLF (SD)	PriMARL -EXP (SD)
video_1	5 (exc.)	97.99 (3.39)	98.69 (1.97)	99.22 (1.44)	99.21 (1.42)	99.27 (1.36)	99.24 (1.43)
	4 (good)	0.15 (0.31)	0.057 (0.15)	0.09 (0.21)	0.11 (0.25)	0.07 (1.16)	0.08 (0.2)
	3 (fair)	0.22 (0.44)	0.11 (0.28)	0.11 (0.27)	0.11 (0.25)	0.13 (0.31)	0.12 (0.28)
	2 (poor)	0.26 (0.56)	0.03 (0.09)	0.06 (0.19)	0.07 (0.16)	0.05 (0.13)	0.07 (0.17)
	1 (bad)	1.39 (2.34)	1.11 (1.88)	0.51 (0.98)	0.51 (0.98)	0.48 (0.93)	0.5 (0.97)
video_2	5 (exc.)	99.11 (1.53)	99.09 (1.41)	99.06 (1.38)	99.08 (1.37)	99.02 (1.47)	99.05 (1.41)
	4 (good)	0.04 (0.12)	0.04 (0.12)	0.15 (0.34)	0.11 (0.29)	0.13 (0.29)	0.11 (0.3)
	3 (fair)	0.09 (0.24)	0.17 (0.32)	0.16 (0.34)	0.19 (0.39)	0.21 (0.42)	0.18 (0.37)
	2 (poor)	0.06 (0.14)	0.05 (0.12)	0.18 (0.42)	0.16 (0.37)	0.17 (0.38)	0.18 (0.4)
	1 (bad)	0.7 (1.22)	0.65 (1.12)	0.45 (0.79)	0.46 (0.81)	0.48 (0.83)	0.48 (0.84)
video_3	5 (exc)	96.82 (3.21)	95.15 (4.68)	99.18 (0.9)	99.2 (0.88)	99.15 (0.92)	99.21 (0.83)
	4 (good)	0.34 (0.42)	0.16 (0.34)	0.14 (0.27)	0.11 (0.26)	0.14 (0.31)	0.1 (0.24)
	3 (fair)	0.25 (0.4)	0.19 (0.4)	0.06 (0.16)	0.07 (0.14)	0.1 (0.22)	0.08 (0.18)
	2 (poor)	0.27 (0.33)	0.34 (0.49)	0.07 (0.15)	0.05 (0.12)	0.06 (0.13)	0.05 (0.13)
	1 (bad)	2.31 (2.66)	4.16 (4.02)	0.55 (0.58)	0.59 (0.62)	0.55 (0.57)	0.56 (0.59)
video_4	5 (exc)	94.42 (3.47)	95.14 (4.62)	98.02 (2.24)	97.8 (2.64)	98.08 (2.12)	98.21 (1.86)
	4 (good)	0.77 (0.84)	0.18 (0.34)	0.35 (0.55)	0.24 (0.4)	0.55 (0.8)	0.3 (0.46)
	3 (fair)	0.59 (0.55)	0.19 (0.37)	0.32 (0.46)	0.26 (0.4)	0.37 (0.6)	0.23 (0.36)
	2 (poor)	0.5 (0.51)	0.31 (0.54)	0.22 (0.42)	0.25 (0.41)	0.21 (0.4)	0.23 (0.43)
	1 (bad)	3.72 (3.37)	4.18 (3.99)	1.08 (1.3)	1.45 (1.97)	0.79 (0.88)	1.03 (1.19)

video_1 and video_2 classes at the 5^{th} PSNR percentiles, we decided to plot the worst percentiles. Depicted in Fig. 7 are the 1^{st} PSNR percentiles averaged over 10 trials for each traffic load. In case of low traffic, the ML-based approaches outperform RADS and FLS in all video classes, except for video_2 where RADS performs slightly better. By assigning MOS levels to the calculated PSNR percentiles, an *excellent* MOS is ensured to all learners by all scheduling approaches in the first two prioritized classes; *good* and *fair* MOS levels are obtained by the ML-based approaches in video_3 and video_4, while *fair* to *bad* levels are obtained through RADS and FLS approaches. In medium traffic load, the best 1^{st} percentiles are obtained by using PriMARL-OPLF in video_2 and video_4 and PriMARL-EXP for the remaining classes. HiMARL provides a balance in PSNR performance within classes, without being the best option in any of them. Correlating to MOS, PriMARL-OPLF can get a *good* level in the video_1 and video_2 classes. However, in the remaining classes, a *bad* MOS level is experienced by all scheduling approaches. When increasing the traffic load to high, *good* MOS levels are obtained only in video_2 class by all ML-based approaches. When looking at the performance of 1^{st} PSNR percentiles for all traffic settings and video classes, the best values are obtained by PriMARL-OPLF and PriMARL-EXP solutions. We can conclude at this point that aiming to maximize the multi-objective function in terms of throughput, delay, and PLR will not guarantee the best performance in terms of worst PSNR percentiles, as we have seen in the case of the HiMARL approach.

2) MOS Analysis: This analysis counts the number of PSNR percentiles which falls in the five MOS levels averaged over ten trials in downlink scheduling. Highlighted in green, we represent the best performance in terms of the highest and lowest number of PSNR percentiles with *excellent* and *bad* MOS, respectively. In Tables I, II, and III we present the MOS analysis in the form of numerical results for each of the scheduler type in low, medium, and high traffic load. The results are averaged over $G = 10$ trials and the Standard Deviation (SD) values are reported in brackets.

When scheduling low traffic load of video_1 (Table I), the PriMARL-OPLF provides the highest number of percentiles in *excellent* MOS and the lowest in *bad* MOS. In

TABLE II
MOS LEVELS FOR MEDIUM TRAFFIC

Traffic	MOS	RADS (SD)	FLS (SD)	HiMARL (SD)	PriMARL -BF (SD)	PriMARL -OPLF (SD)	PriMARL -EXP (SD)
video_1	5 (exc)	91.67 (7.19)	98.76 (1.04)	97.75 (2.93)	98.5 (1.61)	98.32 (2.12)	98.2 (2.44)
	4 (good)	1.28 (1.20)	0.11 (0.23)	0.43 (0.72)	0.15 (0.31)	0.39 (0.7)	0.24 (0.47)
	3 (fair)	1.2 (1.08)	0.08 (0.21)	0.4 (0.61)	0.26 (0.47)	0.27 (0.51)	0.24 (0.41)
	2 (poor)	1.17 (1.15)	0.03 (0.08)	0.34 (0.62)	0.15 (0.28)	0.22 (0.43)	0.21 (0.39)
	1 (bad)	4.69 (4.82)	1.01 (1.0)	1.09 (1.49)	0.95 (1.01)	0.82 (0.91)	1.12 (1.56)
video_2	5 (exc)	98.59 (1.29)	98.69 (0.83)	98.8 (1.27)	97.86 (2.57)	98.91 (1.17)	98.4 (1.68)
	4 (good)	0.12 (0.22)	0.18 (0.34)	0.08 (0.17)	0.09 (0.21)	0.07 (0.17)	0.08 (0.18)
	3 (fair)	0.08 (0.16)	0.17 (0.33)	0.14 (0.29)	0.21 (0.39)	0.09 (0.22)	0.13 (0.26)
	2 (poor)	0.12 (0.25)	0.06 (0.17)	0.16 (0.39)	0.47 (0.87)	0.18 (0.4)	0.35 (0.69)
	1 (bad)	1.1 (0.99)	0.92 (0.76)	0.83 (0.94)	1.38 (1.72)	0.76 (0.85)	1.05 (1.23)
video_3	5 (exc)	78.09 (14.13)	74.11 (13.03)	79.14 (18.74)	88.19 (9.48)	74.05 (19.13)	92.23 (7.58)
	4 (good)	1.23 (1.17)	0.66 (0.65)	6 (6.11)	0.57 (0.61)	10.39 (8.48)	1.19 (1.26)
	3 (fair)	1.21 (1.07)	0.92 (0.77)	4.73 (4.68)	0.63 (0.63)	6.06 (5.12)	0.96 (0.99)
	2 (poor)	1.3 (1.11)	1.01 (0.82)	3.23 (3.49)	0.74 (0.68)	3.41 (3.32)	0.86 (0.95)
	1 (bad)	18.19 (12.12)	23.23 (12.19)	6.94 (7.55)	9.88 (8.41)	6.1 (6.56)	4.78 (4.89)
video_4	5 (exc)	81.23 (10.63)	74.6 (12.42)	61.24 (20.66)	67.92 (14)	55.97 (21.65)	73.67 (14.17)
	4 (good)	0.75 (0.69)	0.6 (0.61)	5.22 (4.35)	1.24 (0.91)	9.41 (8.47)	3.28 (2.12)
	3 (fair)	0.84 (0.71)	0.9 (0.71)	4.9 (3.77)	1.25 (0.84)	7.8 (6.31)	2.71 (1.72)
	2 (poor)	0.96 (0.82)	1.07 (0.86)	4.71 (3.71)	1.61 (1.02)	6.04 (4.66)	2.51 (1.75)
	1 (bad)	16.23 (9.73)	22.84 (11.74)	23.95 (15.5)	27.98 (12.89)	20.79 (15.95)	17.84 (10.42)

TABLE III
MOS LEVELS FOR HIGH TRAFFIC

Traffic	MOS	RADS (SD)	FLS (SD)	HiMARL (SD)	PriMARL -BF (SD)	PriMARL -OPLF (SD)	PriMARL -EXP (SD)
video_1	5 (exc)	72.3 (12.22)	98.54 (0.7)	98.7 (2.43)	98.73 (0.88)	98.83 (1.03)	98.77 (1.42)
	4 (good)	2.9 (1.36)	0.15 (0.26)	0.29 (0.57)	0.09 (0.23)	0.26 (0.54)	0.11 (0.26)
	3 (fair)	3.24 (1.57)	0.09 (0.23)	0.24 (0.48)	0.13 (0.24)	0.13 (0.23)	0.13 (0.28)
	2 (poor)	3.59 (1.68)	0.05 (0.14)	0.21 (0.49)	0.12 (0.29)	0.07 (0.19)	0.14 (0.32)
	1 (bad)	17.96 (10.19)	1.18 (0.73)	1.08 (1.27)	0.93 (0.62)	0.71 (0.54)	0.86 (0.94)
video_2	5 (exc)	96.18 (4.02)	98.76 (0.62)	99.44 (0.6)	99.29 (0.76)	99.45 (0.57)	99.43 (0.58)
	4 (good)	0.51 (0.69)	0.23 (0.42)	0.03 (0.07)	0.04 (0.13)	0.04 (0.11)	0.04 (0.09)
	3 (fair)	0.29 (0.41)	0.14 (0.29)	0.03 (0.09)	0.04 (0.11)	0.04 (0.11)	0.05 (0.15)
	2 (poor)	0.44 (0.63)	0.04 (0.11)	0.04 (0.09)	0.05 (0.14)	0.01 (0.02)	0.04 (0.09)
	1 (bad)	2.58 (3.05)	0.82 (0.59)	0.47 (0.51)	0.59 (0.63)	0.47 (0.5)	0.45 (0.48)
video_3	5 (exc)	38.89 (12.99)	43.16 (11.46)	19.35 (21.62)	55.15 (14.4)	11.85 (13.56)	60.23 (19.17)
	4 (good)	2.81 (1.68)	0.81 (0.64)	12.4 (9.04)	0.94 (0.7)	17.48 (11.53)	4.68 (2.2)
	3 (fair)	2.79 (1.75)	1 (0.74)	15.74 (7.47)	1.25 (0.69)	18.21 (7.43)	4.06 (2.12)
	2 (poor)	2.66 (1.71)	1.52 (0.88)	13.77 (5.87)	1.63 (0.98)	14.67 (5.32)	3.69 (1.89)
	1 (bad)	52.86 (13.1)	53.5 (11.4)	38.74 (21.38)	41.03 (14.23)	37.79 (21.85)	27.34 (15.24)
video_4	5 (exc)	48.8 (13.21)	43.44 (13.25)	5.75 (7.93)	31.55 (11.92)	1.06 (2.28)	26.48 (11.24)
	4 (good)	1.03 (0.71)	0.81 (0.63)	2.15 (2.63)	1.34 (0.8)	2.24 (3.42)	6.19 (1.8)
	3 (fair)	1.4 (0.83)	1.06 (0.69)	3.24 (3.85)	1.69 (0.97)	4.49 (5.96)	5.84 (1.82)
	2 (poor)	1.61 (0.97)	1.38 (0.81)	5.08 (5.01)	2.1 (1.13)	6.59 (7.05)	5.26 (1.66)
	1 (bad)	47.16 (13.54)	53.31 (13.14)	83.79 (17.31)	63.32 (12.25)	85.63 (16.98)	56.21 (12.82)

the second prioritized class, more than 99% of the PSNR percentiles are in *excellent* MOS level for all approaches. The same performance is obtained in video_3 by MARL-based approaches only, while a degradation of more than 2% in *excellent* MOS level is obtained by the other approaches (RADS and FLS). When scheduling learners in video_4, PriMARL-EXP, PriMARL-OPLF and HiMARL achieve a percentage higher than 98% of the PSNR percentiles with *excellent* MOS, while the lowest amount of percentiles in *bad* MOS is obtained when using PriMARL-OPLF. In case of RADS and FLS, more than 3% degradation of *excellent* MOS services can be observed. When looking at the overall performance in low traffic setting, PriMARL-EXP, PriMARL-OPLF, and HiMARL could be identified as the best options.

In medium traffic load (Table II), all candidates except for RADS obtained nearly the same performance of 98% PSNR percentiles with *excellent* MOS when scheduling learners from the video_1 and video_2 classes. For video_3, the PriMARL-EXP solution achieves the highest and lowest amount of percentile with *excellent* and *bad* MOS levels, respectively, placing it as the best option among the candidates. HiMARL follows the PriMARL-OPLF policy by degrading the performance uniformly over the MOS levels. RADS achieves a similar performance in terms of the percentage of PSNR percentiles with *excellent* MOS, but it

substantially increases the amount of percentiles located at the *bad* MOS level. However, being unable to respect the imposed prioritization scheme, RADS provides the highest and lowest number of PSNR percentiles with *excellent* and *bad* MOS respectively, when scheduling learners in the `video_4` class. Looking at the overall MOS performance within the video classes with medium traffic load, it can be concluded that PriMARL-OPLF is the best option for `video_1` and `video_2`, while PriMARL-EXP can achieve a much higher percentage of *excellent* PSNR percentiles when scheduling the `video_3` and `video_4` classes.

By increasing the traffic load from medium to high (Table III), it can be observed that RADS allocates more resources to `video_4` with the lowest priority requirements and degrades the MOS levels in the first prioritized service classes, `video_1` and `video_2`. In these cases, all other scheduling options provide more than 98% of the PSNR percentiles with *excellent* MOS level, of which PriMARL-OPLF is the best option. In `video_3`, PriMARL-EXP outperforms other scheduling candidates by achieving more than 60% of the percentiles in *excellent* MOS and around 27% of the percentiles with *bad* MOS level. The second best option in this case is the PriMARL-BF approach with 55% percentiles in *excellent* MOS and with 41% in *bad* MOS. As previously observed in lower traffic settings, the PriMARL-OPLF scheduling technique aims to minimize the packet loss for all learners without any specific control on PSNR performance. The HiMARL approach follows the OPLF scheduling rule for the resource allocation in `video_3` and increases consistently the percentage of PSNR percentiles in *fair*, *poor* and *bad* MOS levels. When scheduling learners in `video_4`, FLS obtains the same performance as for the `video_3` class, which means that only the group of `video_1` and `video_2` services is prioritized over the `video_3` and `video_4` classes. Looking at the performance among ML-based approaches, PriMARL-BF can get the highest amount of PSNR percentiles with *excellent* MOS of about 32%, while PriMARL-EXP gets the lowest percentage of percentiles with *bad* MOS of about 56%.

Summarizing the results from Tables I, II and III, the following conclusions can be drawn from the perspective of MOS levels over the calculated PSNR percentiles:

a) RADS does not respect the imposed prioritization scheme and provides higher number of PSNR percentiles with *excellent* MOS in the `video_2` and `video_4` classes than in `video_1` and `video_3` respectively, especially for medium and high traffic loads;

b) FLS prioritizes between the group of `video_1` and `video_2` classes and the rest, but it cannot prioritize `video_3` over `video_4` and provides nearly the same distribution of MOS levels in both classes for all traffic settings;

c) HiMARL aims at maximizing the multi-objective reward function in terms of QoS requirements for all learners in all video classes, and thus, degrading the amount of learners experiencing *excellent* MOS levels of video content;

d) PriMARL-BF and PriMARL-OPLF are fair options to learners from all classes regardless of the wireless channel conditions, which is why the higher amount of PSNR percentiles with *bad* MOS is obtained, especially when providing `video_3` and `video_4` services at medium and high traffic loads;

e) being able to properly prioritize and schedule learners based on the highest packet delay, PriMARL-EXP provides the best results by substantially improving over other candidates in terms of percentage of results in the *excellent* MOS category.

G. Additional Results

As we observed, the MARL-based approaches are able to prioritize learners from the considered video classes much better when compared to more conventional scheduling approaches, such as RADS and FLS. When evaluating the QoS performance in Fig. 5, we observe that PriMARL-OPLF and PriMARL-EXP obtain the highest throughput levels (5^{th} percentiles) and lowest rates in packet loss (95^{th} percentiles) in different video classes and traffic settings. The HiMARL meta-scheduler provides the best trade-off between PriMARL-OPLF and PriMARL-EXP in terms of delay, PLR, and throughput because a different scheduling rule is selected to perform the radio resource allocation based on the networking conditions in each class. However, only focusing on improving the QoS performance and ensuring a good trade-off between throughput, delay, and PLR does not guarantee an enhanced performance when measuring the perceived QoE.

When evaluating the PSNR and MOS, we considered three levels in traffic load. For our discussion, we would like to find an approximate average number of learners that can be supported in *excellent* MOS in all video classes with different scheduling approaches. So far, in Tables I–III, we averaged the MOS levels over the number of learners in the intervals of [6, 20], [21, 40], and [41, 60], in low, medium, and high traffic load settings, respectively. Then, we can average over the intervals to get the number of learners supported by each traffic setting and we obtain, 12, 30 and 50 for low, medium, and high traffic load, respectively. With the ratios between video classes introduced in Section V-A, the following averaged numbers of learners in each video class are obtained: *a)* in low traffic load, $U_1 = 2$, $U_2 = 2$, $U_3 = 4$, $U_4 = 4$; *b)* in medium traffic load, $U_1 = 5$, $U_2 = 5$, $U_3 = 10$, $U_4 = 10$; *c)* in high traffic load, $U_1 = 8$, $U_2 = 8$, $U_3 = 17$, $U_4 = 17$. Based on the MOS statistics exposed in Tables I, II, and III, we would like to find next an approximate number of learners experiencing *excellent* MOS of video content in each class when employing the best PriMARL scheduling schemes compared to other approaches.

In low traffic settings (Table I), the thresholds of dropping MOS from *excellent* to lower levels is about 50% for slideshow content with high and low quality (`video_1` and `video_2`), and 75% for animation and screencast contents with low quality (`video_3` and `video_4`). All scheduling approaches analysed in Table I achieve more than 94% of PSNRs in *excellent* MOS, and therefore, all 12 learners from different video classes experience an *excellent* MOS level of the viewed content most of the time.

When scheduling medium traffic load (Table II), we approximate the number of learners to five with *excellent* MOS for

all scheduling approaches when watching slideshow content at high and low quality (`video_1` and `video_2`). In case of `video_3`, PriMARL-EXP and PriMARL-BF provide *excellent* MOS to nine learners when watching animation, while RADS and HiMARL handle eight, and FLS and PriMARL-OPLF seven learners. When scheduling learners with screencast content, eight of them can get *excellent* MOS with RADS, seven with PriMARL-EXP, FLS and PriMARL-BF, and six with HiMARL and PriMARL-OPLF. By summing the number of learners experiencing *excellent* MOS in all video classes, we observe that both RADS and PriMARL-EXP support the same number of learners with this quality, which is 26. However, PriMARL-EXP prioritizes the viewers with animation content (`video_3`) much better compared to the ones with screencast (`video_4`) content ($U_3 : U_4 = 9 : 7$ for PriMARL-EXP compared to $U_3 : U_4 = 8 : 8$ for RADS).

In high traffic load settings (Table III), all eight learners can receive slideshow content at high quality with *excellent* MOS when employing the analysed scheduling approaches, except for RADS which supports only six viewers. At a lower quality of `video_2`, RADS provides nearly the same performance as other candidates supporting the same number of learners with *excellent* MOS level. When delivering animation content (`video_3`), PriMARL-EXP is the best option by supporting ten viewers, followed by PriMARL-BF with nine, RADS and FLS with seven, HiMARL with three, and PriMARL-OLPF with two learners. In case of screencast video streaming and scheduling, eight, seven, five, and five viewers are supported by RADS, FLS, PriMARL-BF and PriMARL-EXP approaches respectively. By summing the number of viewers with *excellent* MOS in all video classes, PriMARL-EXP remains the best option with 31 learners, PriMARL-BF and FLS support 30 learners with the same QoE. However, PriMARL-BF prioritizes animation better compared to screencasts. The list continues with the RADS, HiMARL and PriMARL-OPLF schedulers that can obtain *excellent* MOS for 29, 20, and 18 learners, respectively.

H. Summary

The QoS analysis (Section V-E) shows that, with few exceptions, PriMARL-OPLF achieves the best results when measuring the 5^{th} throughput and 95^{th} PLR percentiles, while PriMARL-EXP performs slightly better when measuring the 95^{th} delay percentile. When monitoring the time when all QoS requirements are met for each video class, PriMARL-EXP performs better than all other candidates, especially in case of medium and higher traffic load when lower prioritized video services are delivered. From the QoE analysis (Section V-F), PriMARL-OPLF gets the highest level of 1^{st} PSNR percentiles in almost all cases. However, when considering the QoE levels for all traffic loads, the PriMARL-EXP obtains the highest number of PSNR percentiles with *excellent* MOS while maintaining the required prioritization between video classes. Further analysis (Section V-G) shows that PriMARL-EXP outperforms other candidates in terms of the number of learners experiencing *excellent* MOS values of video content in each class and traffic load. Compared to the previous work [7], in which HiMARL is proposed to decide at each TTI the prioritization among classes as well as the selection of the scheduling rule for each class, in this paper we show that, from a QoE perspective, maintaining the static scheduling rule in the frequency domain is more efficient.

VI. Conclusion

This paper proposes a PriMARL-based decision-making solution to improve the QoS and QoE provisioning when delivering heterogeneous educational video content in the context of remote education. The proposed PriMARL framework employs an intelligent agent for each class of service that learns to claim its own priority to be scheduled in the frequency domain through a neural network. All agents are cooperating under the form of a joint action to be applied to maximize the overall QoS provision in all classes. Simulation results show that ensuring a good QoS performance does not guarantee excellent QoE levels in different prioritized video classes. We also observed that the scheduling rule which is employed to conduct the scheduling and radio resource allocation plays a crucial role in obtaining high QoE. Among all options analysed in this paper, the proposed PriMARL-based prioritization scheme with exponential scheduling rule works best in terms of perceived QoE. The proposed approach supports 100%, 86%, and 62% of learners with excellent MOS in low, medium, and high traffic settings, respectively.

Acknowledgment

G.-M. Muntean and I. Tal would like to acknowledge the Science Foundation Ireland grant 13/RC/2094_P2 to Lero.

References

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

[2] A. Sepúlveda, "The digital transformation of education: Connecting schools, empowering learners," Broadband Comm. Sustain. Develop. Working Group School Connectivity, Int. Telecommun. Union, Geneva, Switzerland, 2020.

[3] D. Garrison and H. Kanuka, "Blended learning: Uncovering its transformative potential in higher education," *Internet High. Educ.*, vol. 7, no. 2, pp. 95–105, 2004.

[4] D. Prestiadi et al., "The effectiveness of online learning at SIPEJAR using video-based learning media," in *Proc. 1st Int. Conf. Inf. Technol. Educ. (ICITE)*, 2020, pp. 535–540.

[5] E. Beaunoyer, S. Dupéré, and M. J. Guitton, "COVID-19 and digital inequalities: Reciprocal impacts and mitigation strategies," *Comput. Human Behav.*, vol. 111, Oct. 2020, Art. no. 106424.

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

[7] I.-S. Comsa et al., "A machine learning resource allocation solution to improve video quality in remote education," *IEEE Trans. Broadcast.*, vol. 67, no. 3, pp. 664–684, Apr. 2021.

[8] L. Busoniu, R. Babuska, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[9] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[10] M. Sana, A. D. Domenico, W. Yu, Y. Lostanlen, and E. C. Strinati, "Multi-agent reinforcement learning for adaptive user association in dynamic mmWave networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6520–6534, Oct. 2020.
[11] C. Campbell, "Mobile technologies and mobile learning," in *Technology and the Curriculum: Summer*. Sydney, NS, Canada: Power Learn. Solut., ch. 21, 2018.
[12] W. Zhu, X. Wang, and W. Gao, "Multimedia intelligence: When multimedia meets artificial intelligence," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1823–1835, Jul. 2020.
[13] L. Cui, D. Su, S. Yang, Z. Wang, and Z. Ming, "TCLiVi: Transmission control in live video streaming based on deep reinforcement learning," *IEEE Trans. Multimedia*, vol. 23, pp. 651–663, Jan. 2021.
[14] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2017, pp. 197–210.
[15] X. Tan, L. Xu, J. Ni, S. Li, X. Jiang, and Q. Zheng, "Game theory based dynamic adaptive video streaming for multi-client over NDN," *IEEE Trans. Multimedia*, vol. 24, pp. 3491–3505, Jul. 2022.
[16] Z. Chang and S.-H. G. Chan, "An approximation algorithm to maximize user capacity for an auto-scaling VoD system," *IEEE Trans. Multimedia*, vol. 23, pp. 3714–3725, Oct. 2021.
[17] V. Chandrasekhar, Y. Heng, J. Cho, J. Lee, J. Zhang, and J. G. Andrews, "Experience-centric mobile video scheduling through machine learning," *IEEE Access*, vol. 7, pp. 113017–113030, 2019.
[18] P. Semov, P. Koleva, and V. Poulkov, "Adaptive resource scheduling based on neural network and mobile traffic prediction," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, 2019, pp. 585–588.
[19] S.-C. Tseng, Z.-W. Liu, Y.-C. Chou, and C.-W. Huang, "Radio resource scheduling for 5G NR via deep deterministic policy gradient," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2019, pp. 1–6.
[20] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, Aug. 2019.
[21] J. Li and X. Zhang, "Deep reinforcement learning-based joint scheduling of eMBB and URLLC in 5G networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1543–1546, Sep. 2020.
[22] Z. Gu et al., "Knowledge-assisted deep reinforcement learning in 5G scheduler design: From theoretical framework to implementation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2014–2028, Jul. 2021.
[23] S. Mollahasani, M. Erol-Kantarci, M. Hirab, H. Dehghan, and R. Wilson, "Actor-critic learning based QoS-aware scheduler for reconfigurable wireless networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 45–54, Jan./Feb. 2022.
[24] G. Piro, L. Grieco, G. Boggia, R. Fortuna, and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052–1065, Oct. 2011.
[25] 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 *Proc. IEEE Veh. Technol. Conf. (VTC-Spring)*, May 2010, pp. 1–5.
[26] J. V. Den Eynde and C. Blondia, "A minimal delay violation downlink LTE scheduler," in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, 2021, pp. 387–390.
[27] Y. Xing, G. Chuai, W. Gao, and Q. Liu, "A resource scheduling algorithm based on service buffer for LTE-R network," in *Proc. Int. Conf. Commun., Signal Process., Syst.*, 2019, pp. 646–654.
[28] "Technical specification group services and system aspects; policy and charging control architecture release 12, v.12.2.0," 3GPP, Sophia Antipolis, France, Rep. TS 23.203, 2013.
[29] I.-S. Comsa, "Sustainable scheduling policies for radio access networks based On LTE technology," Ph.D. dissertation, School Comput. Sci. Technol., Univ. Bedfordshire, Luton, U.K., 2014.
[30] I.-S. Comsa, 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," *IEEE Trans. Broadcast.*, vol. 67, no. 1, pp. 212–224, Mar. 2021.
[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[32] C. Szepesvari, *Algorithms for Reinforcement Learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning). San Rafael, CA, USA: Morgan Claypool Publ., 2010.
[33] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open-source framework," *IEEE Trans. Veh. Netw.*, vol. 60, no. 2, pp. 498–513, Feb. 2011.
[34] 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.
[35] S.-B. Lee, G.-M. Muntean, and A. F. Smeaton, "Performance-aware replication of distributed pre-recorded IPTV content," *IEEE Trans. Broadcas.*, vol. 55, no. 2, pp. 516–526, Jun. 2009.
[36] A. Moldovan, I. Ghergulescu, and C. H. Muntean, "VQAMap: A novel mechanism for mapping objective video quality metrics to subjective MOS scale," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 610–627, Sep. 2016.

Ioan-Sorin Comşa received the Ph.D. degree from the Institute for Research in Applicable Computing, University of Bedfordshire, U.K., in June 2015. He is a Data Scientist with the Swiss Distance University of Applied Sciences. He was also a Ph.D. Researcher with the Institute of Complex Systems, University of Applied Sciences of Western Switzerland. He worked as a Research Engineer with CEA-LETI, Grenoble, France. Since 2017, he has been a Research Assistant with 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 received the Ph.D. degree in technology enhanced learning from the National College of Ireland. She is an Associate Professor with Swinburne University of Technology, Melbourne, Australia, and an Anna Boyksen Fellow with the Technical University of Munich, Germany. Her research interest includes video-based learning, serious games, and virtual reality. She is a Senior Editor of *Information Technology & People* and on the Editorial Board of the *International Journal of Game-Based Learning*.

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

Christof Imhof received the degree in psychology and the Ph.D. degree in 2022 from the University of Bern. He has been with the Institute for Research in Open, Distance, and elearning since 2016. His research focus lies primarily on procrastination and other types of dilatory behavior in the context of adaptive learning, which also served as the topic of his doctoral thesis. Other research interests include the detection of emotions with objective measures such as eye-tracking combined with emotional word lists.

Per Bergamin is a Professor of Didactics in Distance Education and E-Learning with the Swiss Distance University of Applied Sciences. Since 2006, he acts as the Director of the Institute for Research in Open-, Distance-, and eLearning. In 2020, he was also appointed as an Extraordinary Professor with the Faculty of Education, North-West University, South Africa. From 2016, he holds the UNESCO Chair on personalized and adaptive distance education. His research activities focus on self-regulated and technology-based personalized and adaptive learning. Central aspects are instructional design, usability, and application implementation.

Gabriel-Miro Muntean (Fellow, IEEE) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of the DCU Performance Engineering Lab. He has published over 450 papers in top international journals and conferences, authored four books and 28 book chapters, and edited 9 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 (Member, IEEE) received the Ph.D. degree from Dublin City University, Ireland, in 2005. She is an Associate Professor with the School of Computing, National College of Ireland. She performed various research activities in the past 18 years fostering and promoting research, leading research projects, supervising Ph.D. and M.Sc. students, and publishing over 120 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 received the Ph.D. degree from Dublin City University, Ireland, in 2012. She is a Senior Lecturer with the Design Engineering and Mathematics Department, Middlesex University, London, U.K. She published in prestigious international conferences and journals and has one authored and five edited books. Her research interests include mobile and wireless communications, quality of experience, multimedia streaming, handover and network selection strategies, and digital twin modeling. She is an Associate Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.