

LearnSDN: Optimizing Routing Over Multimedia-based 5G-SDN using Machine Learning

Ahmed Al-Jawad¹, Ioan-Sorin Comşa², Purav Shah¹, Ramona Trestian¹

¹Middlesex University, London, UK

aa3512@live.mdx.ac.uk, {p.shah, r.trestian}@mdx.ac.uk

² Swiss Distance University of Applied Sciences, Brig, Switzerland

ioan-sorin.comsa@ffhs.ch

Abstract—With the advent of 5G networks and beyond, there is an increasing demand to leverage Machine Learning (ML) capabilities and develop new and innovative solutions that could achieve efficient use of network resources and improve users' Quality of Experience (QoE). One of the key enabling technologies for 5G networks is Software Defined Networking (SDN) as it enables fine-grained monitoring and control of the network. Given the variety of dynamic networking conditions within 5G-SDN environments and the diversity of routing algorithms, an intelligent control of these strategies should exist to maximize the Quality of Service (QoS) provisioning of multimedia traffic with more stringent requirements without penalizing the performance of the background traffic. This paper proposes LearnSDN, an innovative ML-based solution that enables QoS provisioning over multimedia-based 5G-SDN environments. LearnSDN uses ML to learn the most convenient routing algorithm to be employed on the background traffic based on the dynamic network conditions in order to cater for the QoS requirements of the multimedia traffic. The performance of the proposed LearnSDN solution is evaluated under a realistic emulation-based SDN environment. The results indicate that LearnSDN outperforms other state-of-the-art solutions in terms of QoS provisioning, PSNR and MOS.

Index Terms—SDN, Reinforcement Learning, QoS, PSNR, Routing Algorithms

I. INTRODUCTION

The advances of technology in the area of Machine Learning (ML), Software Defined Networks (SDN) as well as the rapid deployment of 5G networks are expected to revolutionize the way we communicate, perceive and compute data by enabling a ubiquitous and pervasive paradigm ecosystem [1]. Moreover, the current digital transformation has seen a tremendous increase in the demand for multimedia-rich applications like Augmented Reality (AR) and Virtual Reality (VR). According to Cisco [2] the primary contributors to the global mobile traffic growth are the different mix of wireless devices, including smartphones, machine to machine, tablets, Personal Computers, etc. Moreover, the popularity of Internet of Things (IoT), VR, AR, and ML paves the way for future applications, such as: self driving vehicle diagnostics, Ultra High Definition

(UHD) VR, cloud gaming, etc. However, these multimedia-rich applications will require strict Quality of Service (QoS) requirements that need to be accommodated on various devices characterized by a heterogeneity of hardware platforms and accessed over dynamic network conditions which might hamper their potential [3]. One of the major 5G objectives is to enable the QoS provisioning among three service classes: enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communication (URLLC) and massive Machine Type Communication (mMTC).

The literature provides a wide range of solutions that aim to overcome these challenges and improve QoS, including the use of Multi-Path Transmission Control Protocol (MPTCP) in SDN to improve network resources utilization and the user's QoE for delivering multimedia services over 5G networks [4]. Similarly, Rischke et al. in [5] propose QRSDN, a Reinforcement Learning (RL) routing approach for SDN that enables multi-path routing.

Qadeer et al. [6] explore the use of heuristic RL for flow-level dynamic bandwidth allocation within an SDN-enabled edge cloud system. While Zheng et al. [7] introduce a supervised ML method that classifies the traffic based on specific QoS requirements. The authors propose QAR, a QoS aware routing algorithm that aims to find the most suitable path that minimizes the average link occupation time or maximizes the average path residual capacity. On other hand, Chiu et al. [8] propose RED-STAR, a reinforcement discrete learning service-oriented multipath routing solution for SDN. RED-STAR classifies the service network traffic and makes use of a differentiated reward scheme to dynamically distribute the appropriate routes to the specific service traffic. In our previous work [9], [10], we used RL to select the best routing algorithm for the QoS-based traffic flows only, while the routing strategy for the background traffic flows was set static. However, this approach would actually disturb the users' Quality of Experience (QoE) of QoS-based traffic due to constant interruptions triggered by periodical re-routing decisions.

In contrast to the previous works, this paper introduces LearnSDN and utilizes RL to decide the most appropriate

routing algorithm to be enforced on the background traffic so that the QoS provisioning is maintained for the QoS-based traffic without a significant disruption on the rest of the traffic. Thus, LearnSDN seeks to maximize the return reward of the SDN and maintain the resource reservation of QoS-based traffic, by dynamically selecting a suitable routing algorithm for the background traffic and avoiding interruptions in the QoS-based traffic.

II. LEARNSDN FRAMEWORK

The proposed LearnSDN framework is shown in Figure 1. At user side there are various traffic flows belonging to different 5G service classes, such as: URLLC, eMBB, and mMTC. The 5G-SDN network side consists of the Radio Access Network (RAN) and the transport and core network. The LearnSDN is built on top of the transport/core 5G-SDN and consists of three layers: data, control and application layers. The proposed solution is integrated within the application layer and consists of: **(1) LearnSDN** module - includes the main RL-based solution that decides from a set of centralized routing algorithms like: Minimum Hop Algorithm (MHA) [11], Widest Shortest Path (WSP) [11], Shortest Widest Path (SWP) [11], and Minimum Interference Routing Algorithm (MIRA) [12], the most appropriate routing algorithm to be enforced on the background traffic within SDN; **(2) Routing Manager** module - is triggered by the LearnSDN module to reroute the active background traffic; **(3) SLO Repository** module - maintains the Service Level Objectives (SLO) that describe the service performance in measurable terms (i.e., packet loss, throughput, etc.); **(4) Admission Control** module - decides on accepting or rejecting the incoming traffic requests; **(5) Topology Monitor** module - continuously monitors the SDN state and records any changes in topology for a global view of the network; **(6) Flow Tracker** module - tracks all the flows within SDN, updates the flow states and maintains statistics; and **(7) Active Flows Monitor** module - continuously monitors all the active and inactive flows within SDN.

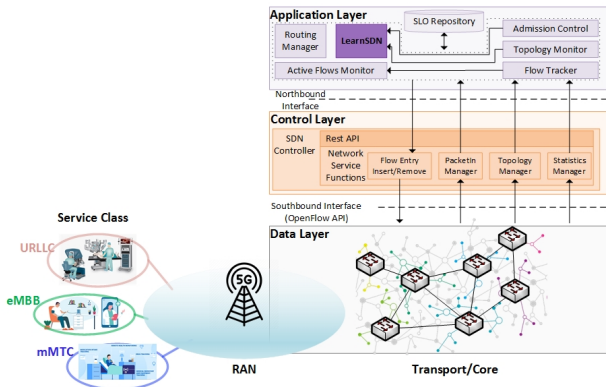


Fig. 1. LearnSDN Framework

III. SYSTEM MODEL

A. Problem Formulation

In general, the topology of any SDN environment in the data plane is modeled by a graph $G(N, L)$. Where N is the set of nodes, with each node representing an SDN switch and L is the set of links that connect any two nodes. Thus, the absolute number of nodes and links within our SDN is equal to $|N|$ and $|L|$. Each link $l \in L$ can be described by a finite capacity C_l and a remaining available bandwidth Bw_l . For the purpose of our problem, we define two types of traffic flow classes, namely F_{QoS} representing the class of QoS-based traffic flows and F_{Bkg} representing the class of background flows. Thus, each traffic flow f in SDN, belongs to a set of flows $F = (F_{QoS} \cup F_{Bkg})$. Additionally, each flow f is further described by the specific traffic type v (i.e., video, HTTP, FTP) and denoted by f_v . Consequently, the remaining available bandwidth Bw_l of link l is determined by $Bw_l = C_l - \sum \rho_f$, where ρ_f represents the total bit rate of flow f . In this context, a feasible path P connecting a source and a destination node, and consisting of a set of links $P = \{l_1, \dots, l_n\}$ needs to be found by a routing algorithm. Thus, the main objective is to enforce the right routing algorithm on the background traffic flows, that maximizes the flows within SDN that satisfy the Service Level Agreement (SLA) requirements in terms of throughput, packet loss and rejection rate. Consequently, the overall optimization problem is formulated below subject to the identified constraints:

$$\text{maximize} \quad \sum_{f \in F} \sum_{p \in P} \Upsilon_{p,f} (\alpha_f \cdot \beta_f \cdot \gamma_f) \quad (1a)$$

$$\text{subject to} \quad \sum_{f \in F} \delta_{l,f} \cdot \rho_f \leq C_l, \quad \forall l \in L, \quad (1b)$$

$$\sum_{p \in P} \Upsilon_{p,f} = 1, \quad \forall f \in F, \quad (1c)$$

$$\sum \alpha_f = 0, \quad \forall f \in F, \quad (1d)$$

$$\sum \beta_f = 0, \quad \forall f \in F, \quad (1e)$$

$$\sum \gamma_f = 0, \quad \forall f \in F, \quad (1f)$$

$$\alpha_f \in \{0, 1\}, \quad \forall f \in F, \quad (1g)$$

$$\beta_f \in \{0, 1\}, \quad \forall f \in F, \quad (1h)$$

$$\gamma_f \in \{0, 1\}, \quad \forall f \in F, \quad (1i)$$

$$\delta_{l,f} \in \{0, 1\}, \quad \forall f \in F, \forall l \in L, \quad (1j)$$

$$\Upsilon_{p,f} \in \{0, 1\}, \quad \forall f \in F, \forall p \in P \quad (1k)$$

where $\Upsilon_{p,f} \in \{0, 1\}$ represents a decision variable that is 0 if flow f does not pass through path p , and 1, otherwise; $\delta_{l,f} \in \{0, 1\}$ represents a decision variable with value 1 indicating that the flow f travels along link l , and 0 otherwise. Constraint (1b) indicates that the sum of all the flows' throughput travelling along link l must not exceed its capacity C_l . Constraint (1c) restricts any flow in SDN to be routed on one path only. The rest of the constraints (1d)-(1k) stipulate that a flow f must satisfy the SLA. Where α_f , β_f , and γ_f representing decision variables

of value 0 if flow f satisfies the SLA in terms of throughput $\Theta_{f,thr}$, packet loss $\Theta_{f,loss}$, and rejection rate $\Theta_{f,rej}$, and 1 otherwise. The defined optimization problem would be similar to our previous work, REDO in [9]. However, the difference appears at the definition of the action space.

B. RL-Based Solution

Q-learning is used to learn the suitable action to be employed at every network state. By employing RL there are three essential elements to be considered, namely the environment, the agent and the action set. The environment is the dynamic network state monitored at discrete times, $t = 0, 1, 2, \dots$ and S_t represents the network state obtained at time t . The agent is the learning entity that will take an action a_t to maximize a defined network reward function, R_t . The action is represented by the routing algorithm to be enforced on the background traffic of the current network state. Based on the reward value R_t that represents the score of applying action a_t in S_t , as SDN move into the next state S_{t+1} the learning entity will tune its action in order to obtain higher rewards under the further states.

1) *state space*: Our main objective is to maintain acceptable QoS levels for the active QoS-based traffic flows f_{QoS} that have stringent QoS requirements. Thus, we define the system state S as follows:

$$S = [\psi, \tau_{QoS}, \mu_{QoS}, \phi_{QoS}] \quad (2)$$

where $\psi \in \{low, medium, high\}$ is the traffic load of SDN. τ_{QoS} and μ_{QoS} are given by eq. (3) and eq. (4), respectively and indicate if the throughput and packet loss rate requirements are met for QoS-based traffic. While ϕ_{QoS} given in eq. (5) indicates if a specific threshold is satisfied by the rejection ratio.

$$\tau_{QoS} = \begin{cases} 1 & \text{if } \sum \alpha_{f_{QoS}} = 0, \\ 0 & \text{if } \sum \alpha_{f_{QoS}} > 0 \end{cases} \quad (3)$$

$$\mu_{QoS} = \begin{cases} 1 & \text{if } \sum \beta_{f_{QoS}} = 0, \\ 0 & \text{if } \sum \beta_{f_{QoS}} > 0 \end{cases} \quad (4)$$

$$\phi_{QoS} = \begin{cases} 1 & \text{if } \sum \gamma_{f_{QoS}} = 0, \\ 0 & \text{if } \sum \gamma_{f_{QoS}} > 0 \end{cases} \quad (5)$$

2) *action space*: LearnSDN defines a set of routing algorithms $O_{Bkg} = \{MHA, WSP, SWP, MIRA\}$. The action will enforce the choice of routing algorithm on the active background traffic flows within the SDN environment. The routing algorithm for the QoS-based traffic flows is fixed and the QoS-based traffic flows are routed using MIRA.

3) *reward function*: As the system is moving from one state to another, an action is executed on each state and a reward is sent back as feedback for each action taken. Thus, the reward is represented by a function that would map the system performance on a certain action taken during a given state into a scalar value. This scalar value will indicate how good was the action applied on that particular state.

We define the total reward as a weighted sum of the overall reward R_v of each traffic class v as follows:

$$R_v = \omega_{thr} * \frac{\sum_{f_v \in F_v} R_{thr,f_v}}{N} + \omega_{loss} * \frac{\sum_{f_v \in F_v} R_{loss,f_v}}{N} + \omega_{rej} * R_{rej,v} \quad (6)$$

where ω_{thr} , ω_{loss} and ω_{rej} represent the throughput, packet loss and rejection rate weights, respectively. For the purpose of this work, $\omega_{thr} = \omega_{loss} = \omega_{rej} = 1/3$. It is also noted that the overall reward R_v for traffic class v consists of three sub-rewards, such that:

(1) R_{thr,f_v} given by eq. (7) describes the variation of the measured throughput $\tilde{\rho}_{f_v}$ of flow f from the corresponding SLA, $\theta_{v,thr} \in \Theta_f$ representing the minimum throughput requirement for flow f in class v . The maximum possible reward of 1 is achieved if the requirements are met.

$$R_{thr,f_v} = \begin{cases} 1 - \left[\frac{\theta_{v,thr} - \tilde{\rho}_{f_v}}{\theta_{v,thr}} \right] & \text{if } \tilde{\rho}_{f_v} \leq \theta_{v,thr} \\ 1 & \text{if } \tilde{\rho}_{f_v} > \theta_{v,thr} \end{cases} \quad (7)$$

(2) R_{loss,f_v} given by eq. (8) describes the variation of the measured packet loss rate $\tilde{\mu}_{f_v}$ of flow f from the corresponding SLA, $\theta_{v,loss} \in \Theta_f$ representing the maximum packet loss requirement.

$$R_{loss,f_v} = \begin{cases} 1 - \left[\frac{\tilde{\mu}_{f_v} - \theta_{v,loss}}{\tilde{\mu}_{f_v}} \right] & \text{if } \tilde{\mu}_{f_v} \geq \theta_{v,loss} \\ 1 & \text{if } \tilde{\mu}_{f_v} < \theta_{v,loss} \end{cases} \quad (8)$$

(3) $R_{rej,v}$ given by eq. (9) describes the variation of the rejection rate $\tilde{\phi}_v$ of a traffic class v from the corresponding SLA, $\theta_{v,rej} \in \Theta_f$ representing the rejection rate requirement.

$$R_{rej,v} = \begin{cases} 1 - \left[\frac{\tilde{\phi}_v - \theta_{v,rej}}{\tilde{\phi}_v} \right] & \text{if } \tilde{\phi}_v \geq \theta_{v,rej} \\ 1 & \text{if } \tilde{\phi}_v < \theta_{v,rej} \end{cases} \quad (9)$$

In the scope of this work, HD video traffic is used to represent the QoS-based traffic class and SD video, HTTP, and FTP traffic are used to represent the background traffic class. In this context, the total reward R is formulated as:

$$R = \underbrace{\omega_{HD_Video} * R_{HD_Video}}_{\text{QoS-based traffic class}} + \underbrace{\omega_{SD_Video} * R_{SD_Video} + \omega_{ftp} * R_{FTP} + \omega_{http} * R_{HTTP}}_{\text{Background traffic class}} \quad (10)$$

where the values for each traffic weights are set to $\omega_{HD_Video} = 0.63$, $\omega_{SD_Video} = 0.19$, $\omega_{ftp} = 0.09$, and $\omega_{http} = 0.09$, so that $\omega_{HD_Video} + \omega_{SD_Video} + \omega_{ftp} + \omega_{http} = 1$. The weights are set based on the predictions forecasted by Cisco in [13]. A total of 60 individual trials were conducted to train the proposed LearnSDN solution where each trial had a total of 1500 seconds run time. In order to ensure

convergence to the optimal policy the discount factor was set to 0.9 while the learning rate was set to 0.01.

IV. EXPERIMENTAL SETUP

LearnSDN was implemented and tested through an experimental setup consisting of: (1) Mininet¹ is used to emulate the SDN data plane; (2) the external OpenFlow controller used is Floodlight² that provides RESTful API and network services; and (3) the application layer consisting of the network management functions for performance evaluation. OpenStack is used to host the experimental setup, where one virtual server is used for the Floodlight controller and the application layer and another virtual server is used to run the Mininet test-bench. Additionally, Open vSwitch³ is used as an SDN software switch.

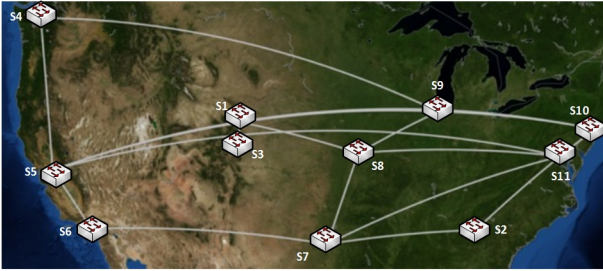


Fig. 2. Implemented Sprint Network Topology

In order to create a more realistic multimedia-based SDN environment we have used the Sprint network topology available from Internet Topology Zoo [14] as depicted in Fig. 2. SDN Openflow switches were used to replace the network nodes, and one host is directly connected to each SDN switch to generate data traffic into the network. In this work, as proof-of-concept, we have used two types of service classes. However, this can be scaled up or down as required. The two service classes are defined as: (1) QoS-based traffic class represented by live HD video streaming with the following characteristics: 1280x720 pixels resolutions, 24 frames per second, average bit rate of 665Kbps, and a total duration of 5 minutes; and (2) the background traffic class represented by buffered SD video streaming, web browsing and file transfer traffic. The characteristics of the SD video traffic are as follows: 640x360 pixels resolution, 24 frames per second, average bit rate of 285kbps, and a total duration of 5 min. The web browsing and file transfer are modeled as HTTP and FTP traffic, respectively [15]. The VLC player tool employed with a Constant Bit Rate (CBR) encoder was used to generate live HD and buffered SD video streaming traffic. FFMPEG video and audio converter⁴ is used to create the video source. The FFMPEG tool, an open-source library, is used to convert between arbitrary sample rates and re-size the audio and video data separately. While Ostinato⁵ traffic

¹Mininet - <http://mininet.org>

²Floodlight - <http://www.projectfloodlight.org>

³ovswitch - <http://openvswitch.org>

⁴FFMPEG tool - <https://ffmpeg.org>

⁵Ostinato - <https://ostinato.org/>

generator tool was used to generate HTTP and FTP traffic. This enabled us to evaluate different traffic mix with 63% live HD video, 19% buffered SD video, 9% HTTP and 9% FTP as per [13] and three different load levels on the network. The load levels were set to 0.5 representing *low load*, 0.75 representing *medium load*, and 1.0 representing *high load*. The network load NL is calculated using (11) based on the link load LL , link capacity C_l and the absolute number of links within the network $|L|$.

$$NL = \frac{\sum_i^{|L|} \frac{LL_i}{C_{li}}}{|L|} \quad (11)$$

The performance of LearnSDN is compared against other state-of-the-art routing algorithms from the literature, such as: MHA, WSP, SWP, MIRA and our previous work, REDO [9]. The comparison is done in terms of throughput, packet loss rate, rejection rate, PSNR and Mean Opinion Score (MOS). The PSNR to MOS mapping is done as per [16] and Table I lists the QoS requirements for each traffic class.

TABLE I
QoS REQUIREMENTS

QoS-based Traffic Class	$\theta_{QoS,thr}$	$\theta_{QoS,loss}$	$\theta_{QoS,rej}$
Live HD video	658 Kb/s	1%	25%
Background Traffic Class	$\theta_{Bkg,thr}$	$\theta_{Bkg,loss}$	$\theta_{Bkg,rej}$
Buffered SD video	279 Kb/s	2%	35%
Web browsing	14 Kb/s	0%	35%
File transfer	180 Kb/s	0%	35%

V. RESULTS AND DISCUSSIONS

The results were averaged over five simulation runs per scenario, with each scenario having a duration of 1500 seconds. The same experiment conditions were kept for each scenario when comparing the different solutions.

Figures 3 and 4 illustrate the throughput and packet loss results for each traffic type under different loads and for each scheme. It can be noted that LearnSDN outperforms the other conventional routing algorithms like MHA, WSP, SWP and MIRA. When compared to REDO, we can notice that LearnSDN will prioritise the HD Video with an increase in throughput. For example, under medium load LearnSDN achieves up to 1.3% increase in throughput for the HD video as compared to REDO. Moreover, the results show that both LearnSDN and REDO exhibit similar lower packet loss under various network loads as compared to the conventional routing algorithms. For example, under high load, LearnSDN decreases the packet loss down to 0.38% for the HD video, REDO goes down to 0.5% packet loss rate, while the conventional routing algorithms exhibit a packet loss rate of more than 5%. Table II illustrates the estimated PSNR and MOS. It can be noticed that REDO and LearnSDN achieve an *Excellent* user perceived quality level for both HD and SD video traffic under low, medium and high traffic loads without penalizing the other traffic types. For example, in case of LearnSDN and REDO, when there is

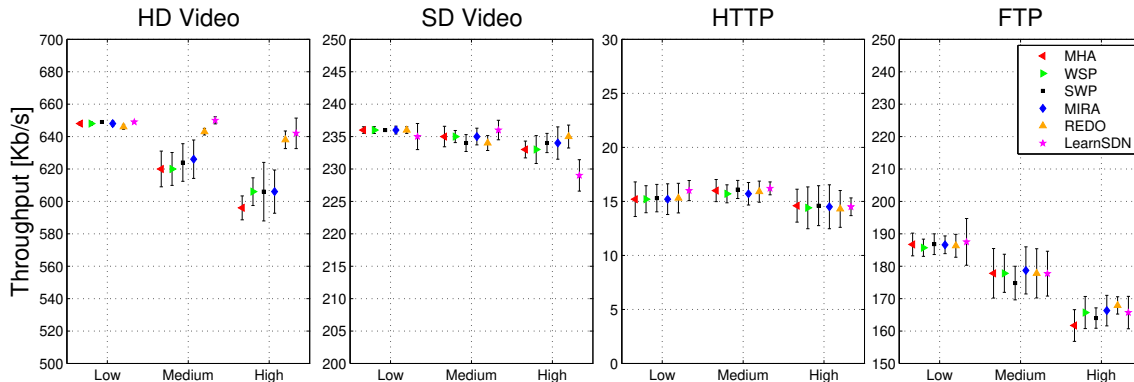


Fig. 3. Throughput results for each traffic type considering different traffic loads

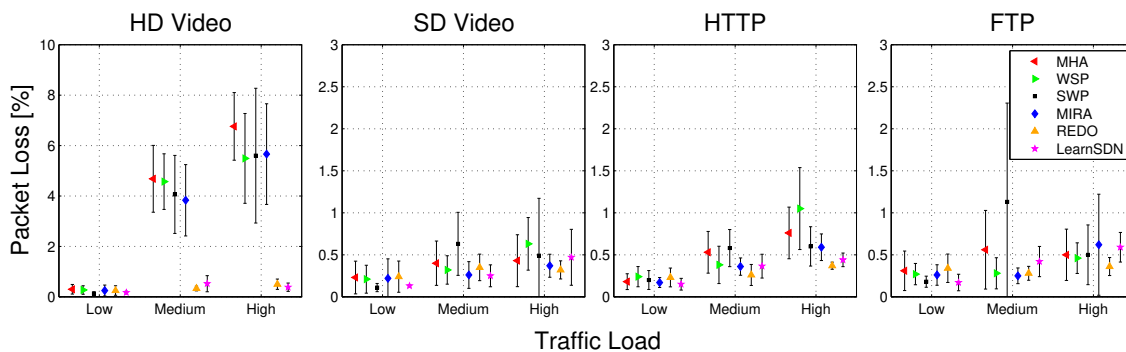


Fig. 4. Packet loss results for each traffic type considering different traffic loads

an increase in the network load from low to high, the user perceived QoE stays *Excellent*. While the user perceived QoE for the conventional routing algorithms, MHA, WSP, SWP, and MIRA decreases from *Excellent* to *Poor*. Consequently, it can be seen that the classical routing algorithms will sacrifice the users' perceived QoE level for the QoS-based traffic class along with penalizing the performance of the other traffic classes in the network, just to accommodate more QoS-based flows.

As explained previously, LearnSDN prioritizes the HD video traffic by using a fixed routing algorithm like MIRA while the background traffic is re-routed using the RL approach based on the dynamic network conditions. In contrast, REDO would re-route the HD video traffic while maintaining a fixed routing for the background traffic using MIRA. However, re-routing of the HD video traffic flows creates interruptions in the video flow that could lead to re-buffering periods. Tan et al. [17] studied the impact of the re-buffering periods on the user perceived QoE and concluded that the MOS will decrease with the increase in buffering percentage level. Even though the buffering has a significant impact on the quality degradation as perceived by the user, this effect is not integrated into the video quality assessment solutions, like PSNR. Consequently, even though LearnSDN and REDO might exhibit similar results

in terms of MOS, the re-buffering effect might decrease the MOS further in case of REDO.

Moreover, Figure 5 illustrates the number of rejected flows under different traffic types as a function of the network traffic load. It can be seen that under highly loaded network, the rejection rate of all solutions increases considerably. However, under medium traffic load, it can be noted that LearnSDN rejects a less number of traffic flows overall, as compared to REDO. Even if under high traffic load, LearnSDN will reject more flows overall, we can notice that it is rejecting less HD video traffic flows as compared to REDO. Consequently, LearnSDN outperforms the other state-of-the-art conventional routing algorithms while offering a better prioritization for HD video flows as compared to REDO.

VI. CONCLUSIONS

With QoE gaining strong momentum due to the ever increasing users' QoE expectations of rich multimedia applications, the focus is now on proposing innovative solutions that leverage the ML capabilities to enable QoE when delivering video content over 5G networks and beyond. In this context, the integration of SDN is seen as an enabling solution for QoE provisioning over 5G-SDN environments especially as QoE is seen to become the biggest differentiator between network operators. In this paper,

TABLE II
AVERAGED ESTIMATED PSNR AND MOS CONSIDERING DIFFERENT TRAFFIC LOADS AND FOR EACH SCHEME

		MHA			WSP			SWP			MIRA		
		low	medium	high	low	medium	high	low	medium	high	low	medium	high
HD	PSNR [dB]	50.5	26.6	23.4	51.4	26.8	25.2	57.7	27.8	25	52	28.3	24.9
	MOS	Exc.	Poor	Poor	Exc.	Poor	Poor	Exc.	Fair	Poor	Exc.	Fair	Poor
SD	PSNR [dB]	52.7	47.9	47.3	53.5	49.8	44	59.1	44	46.1	53.1	51.7	48.6
	MOS	Exc	Exc.	Exc.	Exc.	Exc.	Good	Exc.	Good	Exc.	Exc.	Exc.	Exc.

		REDO			LearnSDN		
		low	medium	high	low	medium	high
HD	PSNR [dB]	52	49.6	46	58	45.7	48.4
	MOS	Exc.	Exc.	Exc.	Exc.	Exc.	Exc.
SD	PSNR [dB]	52.3	49.1	49.8	55.3	52	46.5
	MOS	Exc.	Exc.	Exc.	Exc.	Exc.	Exc.

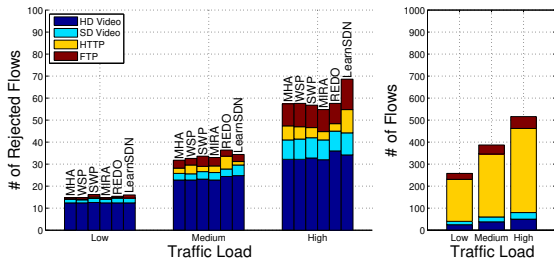


Fig. 5. Number of rejected flows and the total number of flows

we propose LearnSDN, an innovative ML-based solution that would learn the most suitable routing algorithm to be employed on the background traffic, considering the dynamic networking conditions in order to ensure QoS provisioning for QoS-based traffic. LearnSDN was evaluated under a realistic SDN-based experimental setup consisting of Mininet, Floodlight controller and Open vSwitch switches and under dynamic network conditions. The results show that, on average LearnSDN performs better when compared to other state-of-the-art conventional routing algorithms, and finds the most appropriate trade-off between throughput, packet loss rate and rejection rate for the QoS-based traffic class without a significant impact on the other traffic.

REFERENCES

- [1] G. Kakkavas, A. Stamou, V. Karyotis, and S. Papavassiliou, "Network tomography for efficient monitoring in sdn-enabled 5g networks and beyond: Challenges and opportunities," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 70–76, 2021.
- [2] V. Cisco *et al.*, "Cisco visual networking index: Forecast and trends 2018-2023 white paper," 2019.
- [3] R. Trestian, I.-S. Comşa, and M. F. Tuysuz, "Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?" *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 945–977, 2018.
- [4] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifechor, "A novel qoe-centric sdn-based multipath routing approach for multimedia

- services over 5g networks," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.
- [5] J. Rischke, P. Sossalla, H. Salah, F. H. P. Fitzek, and M. Reisslein, "Qr-sdn: Towards reinforcement learning states, actions, and rewards for direct flow routing in software-defined networks," *IEEE Access*, vol. 8, pp. 174 773–174 791, 2020.
- [6] A. Qadeer, M. J. Lee, and K. Tsukamoto, "Flow-level dynamic bandwidth allocation in sdn-enabled edge cloud using heuristic reinforcement learning," in *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2021, pp. 1–10.
- [7] W. Zheng, M. Yang, C. Zhang, Y. Zheng, Y. Wu, Y. Zhang, and J. Li, "Application-aware qos routing in sdn using machine learning techniques," *Peer-to-Peer Networking and Applications*, pp. 1–20, 2021.
- [8] K.-C. Chiu, C.-C. Liu, and L.-D. Chou, "Reinforcement learning-based service-oriented dynamic multipath routing in sdn," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.
- [9] A. Al-Jawad, I.-S. Comşa, P. Shah, O. Gemikonakli, and R. Trestian, "REDO: a reinforcement learning-based dynamic routing algorithm selection method for SDN," in *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2021, pp. 54–59.
- [10] A. Al-Jawad, I.-S. Comşa, P. Shah, O. Gemikonakli, and R. Trestian, "An innovative reinforcement learning-based framework for quality of service provisioning over multimedia-based sdn environments," *IEEE Transactions on Broadcasting*, vol. 67, no. 4, pp. 851–867, 2021.
- [11] Q. Ma and P. Steenkiste, "Quality-of-service routing for traffic with performance guarantees," in *Building QoS into Distributed Systems*. Springer, 1997, pp. 115–126.
- [12] M. Kodialam and T. Lakshman, "Minimum interference routing with applications to mpls traffic engineering," in *Proceedings IEEE INFOCOM Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2000, pp. 884–893.
- [13] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, p. 1, 2018.
- [14] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.
- [15] V. Deart, V. Mankov, and A. Pilugin, "HTTP Traffic Measurements on Access Networks, Analysis of Results and Simulation," in *Smart Spaces and Next Generation Wired/Wireless Networking*. Springer, 2009, pp. 180–190.
- [16] T. Zinner, O. Abboud, O. Hohlfeld, T. Hossfeld, and P. Tran-Gia, "Towards QoE Management for Scalable Video Streaming," in *21th ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE*, Miyazaki, Jap, 3 2010.
- [17] X. Tan, J. Gustafsson, and G. Heikkilä, "Perceived video streaming quality under initial buffering and rebuffering degradations," in *MESAQIN Conference (June 2006)*, vol. 90, 2006.