Middlesex University London

# Middlesex University Research Repository

An open access repository of

Middlesex University research

http://eprints.mdx.ac.uk

Al-Jawad, Ahmed, Shah, Purav ORCID logoORCID: https://orcid.org/0000-0002-0113-5690, Gemikonakli, Orhan ORCID logoORCID: https://orcid.org/0000-0002-0513-1128 and Trestian, Ramona ORCID logoORCID: https://orcid.org/0000-0003-3315-3081 (2018) LearnQoS: a learning approach for optimizing QoS over multimedia-based SDNs. 2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB). In: IEEE BMSB 2018, 06-08 June 2018, Valencia, Spain. ISBN 9781538647295. ISSN 2155-5052 [Conference or Workshop Item] (doi:10.1109/BMSB.2018.8436781)

Final accepted version (with author's formatting)

This version is available at: https://eprints.mdx.ac.uk/24193/

**Copyright:**

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this work are retained by the author and/or other copyright owners unless otherwise stated. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge.

Works, including theses and research projects, may not be reproduced in any format or medium, or extensive quotations taken from them, or their content changed in any way, without first obtaining permission in writing from the copyright holder(s). They may not be sold or exploited commercially in any format or medium without the prior written permission of the copyright holder(s).

Full bibliographic details must be given when referring to, or quoting from full items including the author's name, the title of the work, publication details where relevant (place, publisher, date), pagination, and for theses or dissertations the awarding institution, the degree type awarded, and the date of the award.

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:

eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

See also repository copyright: re-use policy: http://eprints.mdx.ac.uk/policies.html#copy

# LearnQoS: A Learning Approach for Optimizing QoS over Multimedia-based SDNs

Ahmed Al-Jawad, Purav Shah, Orhan Gemikonakli and Ramona Trestian
Design Engineering and Mathematics Department,
Faculty of Science and Technology,
Middlesex University, London, UK
E-mails: AA3512@live.mdx.ac.uk, {p.shah, o.gemikonakli, r.trestian}@mdx.ac.uk

***Abstract*—As video-based services become an integral part of the end-users' lives, there is an imminent need for increase in the backhaul capacity and resource management efficiency to enable a highly enhanced multimedia experience to the end-users. The next-generation networking paradigm offers wide advantages over the traditional networks through simplifying the management layer, especially with the adoption of Software Defined Networks (SDN). However, enabling Quality of Service (QoS) provisioning still remains a challenge that needs to be optimized especially for multimedia-based applications. In this paper, we propose LearnQoS, an intelligent QoS management framework for multimedia-based SDNs. LearnQoS employs a policy-based network management (PBNM) to ensure the compliance of QoS requirements and optimizes the operation of PBNM through Reinforcement Learning (RL). The proposed LearnQoS framework is implemented and evaluated under an experimental setup environment and compared with the default SDN operation in terms of PSNR, MOS, throughput and packet loss.**

***Index Terms*—Software Defined Networks, Quality of Service, Reinforcement Learning.**

## I. Introduction

The continuous growth of video traffic in both fixed and mobile contexts puts significant pressure on the underlying networks and on the service providers that need to enable the provisioning of high Quality of Service (QoS) levels to the end-users. To cope with this explosion of data traffic, the network operators are trying to find new solutions to leverage of their existing network infrastructures and enable efficient resource management while ensuring QoS provisioning to their customers. Expanding the system capacity is not enough without considering the provisioning of Quality of Experience (QoE) to the end users as the basis for network control [1].

One promising solution is the adoption of Software Defined Networking (SDN). With the advent of SDN architecture, the control and forwarding planes become physically and logically separated and a new standardized communication protocol, OpenFlow [2] is introduced. In this way, the SDN controller becomes the main element to maintain and manage the network services, for instance the assurance of QoS.

Moreover, the underlying network technologies need to be assisted by intelligent decision making solutions to enable the efficient resource management. To this end, one of the most promising enablers for the future 5G technologies is Machine Learning. The use of machine learning techniques is currently gaining strong momentum especially when solving various optimization and control problems including resource allocation in 5G [3] or QoE provisioning for video transmissions [4]. Machine learning consists of a set of algorithms that could be used to learn from existing data and make predictions or decisions on the data, enabling intelligent adaptive environments.

This paper proposes **LearnQoS**, a framework that integrates the use of machine learning, policy-based network management (PBNM) and SDN for QoS optimization. LearnQoS makes use of PBNM to facilitate the management of policy violation and traffic engineering over multimedia-based SDNs. Low-level policies are used to define the QoS within the network. LearnQoS, employs Reinforcement Learning (RL) within the decision making and policy violation detection mechanism inside the PBNM, to optimize the network performance by finding the optimal action.

Thus, the LearnQoS policy management consists of the following entities: *(1) Policy Information Point (PIP)* - contains the policy database; *(2) Policy Decision Point (PDP)* - represents the intelligent network element that makes decisions based on policies and network state; *(3) Policy Enforcement Point (PEP)* - enforces the low-level policy rules on the network end-points. RL, on the other hand, consists of three main elements: reward, action, and state. Within a network, quality performance metrics such as: throughput, latency, packet loss, etc. can represent the reward function. While the action space contains possible actions, for instance rerouting and/or rate shaping/limiting/queuing, that can be used to retain a certain policy requirement. The state represents the network state which is monitored steadily by the SDN controller. By integrating RL in the PDP component, the intelligence reacts proactively on inadequate network changes by finding the optimal selection of action set to maximize the long term reward.

Experimental results shows that the proposed solution minimizes the QoS policy violation. As a result of this, the approach reduces the risk of network congestion and effectively utilizes available network resources in the upcoming time window.

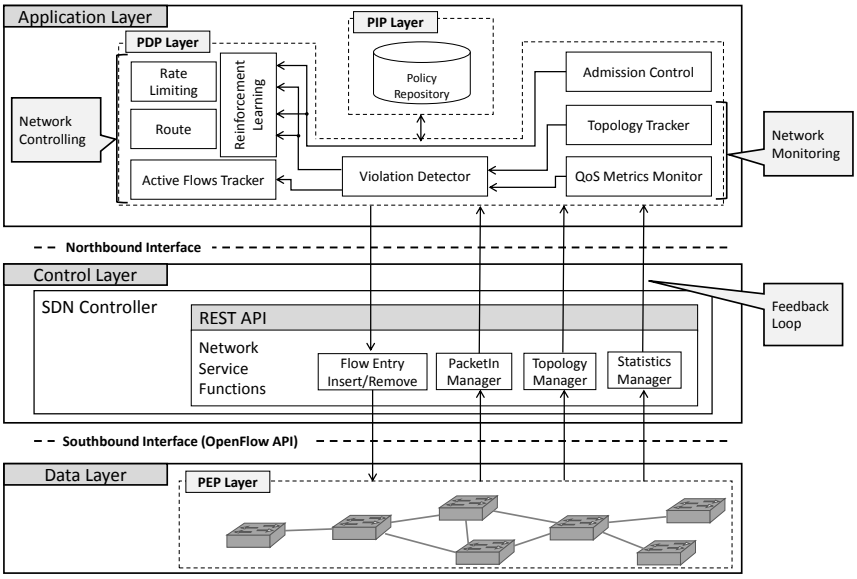


Fig. 1. Proposed LearnQoS Framework

## II. RELATED WORKS

Machine Learning is seen as a promising solution for next generation networks, that will enable intelligent adaptive learning and prediction or decision making so that the applications' QoS and users' requirements can be highly satisfied. The study in [5] presents a survey on the application of AI techniques over the SDN paradigm to solve problems like load balancing and security. The study shows that the integration of AI techniques within SDN is promising, with several research groups introducing the use of reinforcement routing over SDN-based network. The research in [6] introduces a routing algorithm based on RL. The method determines when is the right time to reroute the traffic in order to minimize the packet loss. Similarly, Sendra et al. [7] propose an intelligent routing protocol for SDN based on RL. Whereas, Lin et al. [8] introduces QoS-aware adaptive routing with the aid of RL in a multi-layer hierarchical SDN environment. Kucminski et al. [9] propose a simple routing algorithm for QoS provisioning over SDN-based networks with the proposed solution being evaluated under a real experimental setup. The work in [10] proposes a compression method to reduce the size of data transfer over the control path, while it increases the network observability. The sparsity approximation algorithms are utilized to compress the aggregated data in the SDN switch, while the SDN controller recovers the sparse data.

In our previous work, [11] we proposed a probabilistic QoS routing schemes which makes use of Bayes' theorem and Bayesian network model to determine the link probability to select the route that satisfies the given bandwidth constraint. In general, the global knowledge of the network state information contains uncertainties caused by various perturbation factors. Therefore, the employment of probabilistic schemes into QoS routing shall be favorable to the traditional cost-optimization solutions due to the high control overhead introduced by monitoring the network frequently.

In this work, we propose an enhanced violation detection and intelligent decision making over action set selection within the policy-management framework. The proposed LearnQoS system learns over time and with the help of system modeling it determines the best action to perform for ensuring QoS delivery for multimedia applications. The focus of this research is mainly on handling the decision making inside the policy management framework, where the policy is defined explicitly in a high level definition and translation, carried out on the SDN data plane.

## III. LEARNQOS FRAMEWORK ARCHITECTURE

This section presents the proposed LearnQoS framework integrating the use of RL and PBNM over multimedia-based SDN environments.

### A. LearnQoS Architecture

The proposed PBNM-based SDN framework is illustrated in Figure 1 and consists of the following main components: **(1) Policy Repository** - stores the entire high-level policy rules that reflect the requirements for the agreed services between the service provider and customers. **(2) Topology Tracker** - maps the physical network elements into a graphical structure. The output of this component is fed to the QoS metrics monitoring component to build a global image of the instantaneous network state. **(3) Admission Control** - accepts or declines network connections based on the availability of network resources. **(4) QoS Metrics Monitor** - measures the service quality by periodically sending flow statistics query messages to the switches. The constructed view on the network load is used later by the violation detector to indicate the misbehaving traffic flows.

The units monitors the flow throughput utilization, packet loss percentage and delay for a given flow. **(5) Violation Detector** - represents the validation engine to release the necessary measures in order to converge the network to the state agreed by SLO requirements. This being said, this component is responsible for observing whether a policy is achieved or violated and for finding the congested segment in the network that causing the violation. First, the violation detector examines steadily the validity of SLO policy rules against the real traffic carried over the operational network. Secondly if a violation is detected, it identifies the segment that caused the violation. Finally, the violation detector triggers the type of action to resolve the network bottleneck. **(6) Active Flows Tracker** - it tracks the active flow routes in SDN. The built routing table is utilized by the monitor unit to estimate the throughput per active flow. **(7) Route** - sets up the flow the along the best-effort route. **(8) Rate Limiting** - configures the rate limit parameter along the best-effort route. **(9) Reinforcement Learning** - trains the agent using the earned reward to choose an action (reroute or rate limiting) to be performed on the network.

### *B. Reinforcement Learning-Based Algorithm*

RL is used to train the agent to choose an action through trial and error interaction with an unknown dynamic environment [12]. Initially, the agent observes the state of the environment and chooses the proper action. Based on the agent interaction, the environment returns a numerical reward accordingly, which the agent aims to maximize.

In this work, Q-learning method is employed as an RL technique in order to find an optimal action-selection policy [12]. The Q-learning method uses a Q-table to map the state-action pairs. Q-table represents a table of values for every state and actions possible within the environment. The Q-table is updated according to the following function:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\,[r + \lambda max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)] \quad (1)$$

where $Q$ represents the Q value of a state-action pair. Let $s_t$ and $a_t$ denote the state and the action, respectively, executed by an agent at a time instant t. $r$ is the reward earned from the environment. While $\lambda$ and $\alpha$ represent the discounting rate and the learning rate respectively, with values between 0 and 1.

RL has a trade-off between exploration and exploitation. Exploration is essential to explore actions other than the best candidate. However, it can decrease the network performance due to the randomness. On the other hand, exploitation takes the best decision but other unvisited action may perform better. In this work, $\epsilon$-greedy algorithm is used to give a chance to execute random action. In general using the Q-learning involves two phases: (1) the training phase where the training data set is used to adjust the Q-value in the Q-table, and (2) the testing phase where the data set is used to validate the RL model.

The RL is modeled by defining three elements: state, action, and reward. In our problem, the state is represented by the traffic matrix, while action is defined in Table I and the reward is based on the SLA requirements.

TABLE I
POSSIBLE ACTIONS TAKEN BY RL

| Action | Action Definition |
|---|---|
| do nothing | no action is taken |
| reduce data rate | reduce the rate of best effort flows |
| increase data rate | increase the rate of best effort flows |
| reroute | reroute the best effort flows on alternative path |

## IV. EXPERIMENTAL SETUP

This section presents the experimental setup of the proposed solution.

### *A. Experimental Testbed Setup*

The proposed LearnQoS framework was implemented and tested under the experimental setup illustrated in Fig. 2. The testbed consists of three main elements: (i) Mininet [13] - used to emulate the SDN data plane; (ii) external Floodlight OpenFlow controller [14] - provides RESTful API and network services like the flow entry update; and (iii) the LearnQoS application layer - containing the decision making for QoS policy configurations. The SDN controller and the entire LearnQoS application run on a computer and they are connected via a physical Ethernet link to other computer hosting Mininet. Open vSwitch [15] is used as a software OpenFlow 1.3 enabled switch.

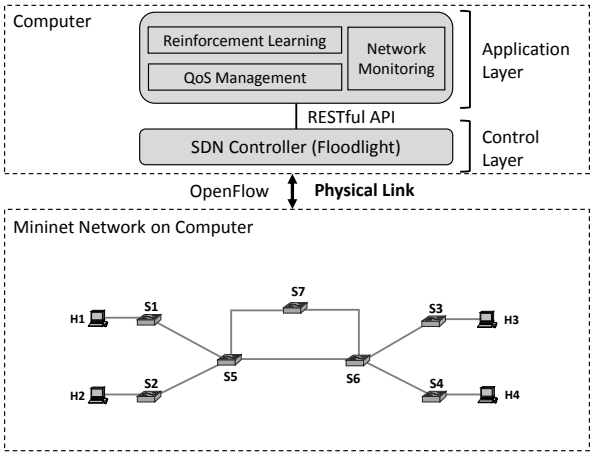


Fig. 2. Experimental Setup for Proof-of-Concept

In order to evaluate the proposed approach, the network topology depicted in Fig. 2 was used for the experimental setup. For the purpose of this experimental test-bed, a simple topology was used because the Q-table approach does not scale easily in a large scale network. As the network becomes larger, the states in the Q-table can easily present an exponentially increase. Thus, as a proof-of-concept, our approach is validated on a small scale network. However, because of the processing capacity limitations of the experimental setup, each link in the topology operates

at the rate of 1 Mb/s. This does not affect the evaluation performance and the approach can be scaled up to a larger network.

TABLE II
PARAMETERS OF TRAFFIC MODELING AND SETUP

| Parameters | Value |
|---|---|
| Video bit-rate | 563 kbps |
| Video frame rate | 24 fps |
| Video duration for QoS traffic | 10 minutes |
| Video duration for best-effort traffic | 2 minutes |
| Experiment duration | 30 minutes |
| Traffic mix | Video = 80%<br>HTTP = 20% |

Video streaming traffic is generated using the VLC player, while background traffic like HTTP is generated using Ostinato [16], a network traffic generator tool. In this way, it is possible to evaluate different traffic mix and load on the network. The traffic generated within the experimental setup contains guaranteed traffic such as video streaming and best-effort traffic represented by video and web flows used as background traffic. Table II illustrates the parameters used for video traffic, the experiment duration as well as the traffic mix. The traffic mix ratio is determined based on the statistics provided by Cisco [17] such that 80% of the total traffic is represented by video traffic and the remaining 20% is represented by HTTP traffic. The parameters for the HTTP traffic model [18] used are listed in Table III. The HTTP traffic is modeled as ON/OFF periods, with the ON period corresponding to the transmission time and the OFF period corresponding to the packet inter-arrival time. For each traffic request, the source and destination host pairs are selected randomly following a uniform distribution.

TABLE III
MODEL PARAMETERS OF WEB TRAFFIC

| Parameters | Best-fit Distribution | Mean & Std. Deviation |
|---|---|---|
| Main object size | Truncated Lognormal | Mean = 10710 bytes<br>Std. dev. = 25932 bytes |
| Embedded object size | Truncated Lognormal | Mean = 7758 bytes<br>Std. dev. = 126168 bytes |
| Number of embedded objects per page | Truncated Pareto | Mean = 5.64<br>Max. = 53 |
| Reading time | Exponential | Mean = 30 sec |
| Parsing time | Exponential | Mean = 0.13 sec |

The following SLO policy is defined for the case-study: *the QoS policy defines that all flows directed from the source to the destination should receive a defined minimum bandwidth and minimal packet loss rate.*

### B. Experimental Scenarios

In order to evaluate the proposed reinforcement learning-based algorithm under dynamic network conditions and policy violations, a scenario with a mix of QoS and best-effort flows is considered. The proposed LearnQoS framework integrates a Q-learning method that triggers the rerouting and rate limiting when the policy violation is happening. For the purpose of this performance evaluation, we define the QoS policy rule for this scenario as: the QoS video traffic from source Host 1 (H1), as indicated in Fig. 2, directed to the destination Host 3 (H3) has a minimum bandwidth threshold requirement of 600 Kb/s and the maximum threshold for packet loss rate is set to 2%. The characteristics of the QoS video traffic are listed in Table II. For the background traffic, a mix of video and HTTP traffic as best-effort are generated between Host 2 (H2) and Host 4 (H4) maintaining the 80% to 20% ratio according to [17]. The monitoring update interval for the proposed LearnQoS framework was set to 1 second.

As the scope of this study does not cover the translation and verification between SLA and SLO levels, the SLO requirement is defined directly without deriving it from the SLA, as follows: The multimedia traffic between the source host, H1 and destination host, H3 satisfies at least the minimum bandwidth requirements (600Kb/s) and minimal packet loss rate below 2%.

### C. Performance Metrics

The following performance metrics are used to assess the performance of the proposed LearnQoS framework in terms of user perceived QoE for video streaming: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM).

A mapping of PSNR and SSIM to the nominal Mean Opinion Score (MOS) is given in Table IV [19]. MOS represents a five point scale used to subjectively assess the users' QoE.

TABLE IV
PSNR AND SSIM TO MOS MAPPING [19]

| MOS | PSNR | SSIM |
|---|---|---|
| 5 (Excellent) | $\geq 45$ | $\geq 0.99$ |
| 4 (Good) | $\geq 33$ & $< 45$ | $\geq 0.95$ & $< 0.99$ |
| 3 (Fair) | $\geq 27.4$ & $< 33$ | $\geq 0.88$ & $< 0.95$ |
| 2 (Poor) | $\geq 18.7$ & $< 27.4$ | $\geq 0.5$ & $< 0.88$ |
| 1 (Bad) | $< 18.7$ | $< 0.5$ |

## V. RESULTS AND DISCUSSIONS

For the performance evaluation, we compare the performance of the proposed LearnQoS framework against the default configuration of the SDN-based network, which computes the shortest path for all traffic types. The comparison is performed on the same random seed to reproduce a deterministic trail. Each experiment is repeated three times and the average outcome are evaluated. The performance evaluation is done in terms of Throughput, Packet Loss Rate, Latency, PSNR, SSIM and MOS of the QoS video flow.

For the evaluation purpose, the data set is divided into two groups: (1) the training data for establishing the Q-table, and (2) the testing data for system performance evaluation. The training phase is performed through the run of several iterations of the RL algorithm.

Figure 3 illustrates the throughput, packet loss rate and latency measurements for the QoS video flow under LearnQoS and the default SDN. Initially, in the route setup phase,

the route manager selects the least loaded path (S1-S5-S6-S3) for the QoS video traffic between H1 and H3, while the best-effort uses the same path (S2-S5-S6-S4) between H2 and H4. Therefore, the link (S5-S6) becomes a shared resource between the QoS traffic and the best-effort traffic.

The results show that LearnQoS identifies the time when the packet loss rate exceeded the limit of 2% imposed by the QoS policy rule. Due to the link congestion, the algorithm reroutes the best-effort flow on an alternative path (S2-S5-S7-S6-S4). On the other hand, the results show that LearnQoS identifies that the load on the S5-S6 segment became low during the experiment run. Hence, LearnQoS redirects the best-effort flows on the (S2-S5-S6-S4) path. However, it identifies later some policy violations at time-stamps 10, 86, 134, 256, 341, and 510 when the packet loss exceeds the limit of 2%. To avoid the congestion, LearnQoS choses the action of rate reduction to decrease the best-effort traffic load. However, due to the high volume of traffic this action does not help so LearnQoS reroutes the best-effort traffic on the other alternative path (S2-S5-S7-S6-S4) and increases its traffic rate. Table V lists the average PSNR

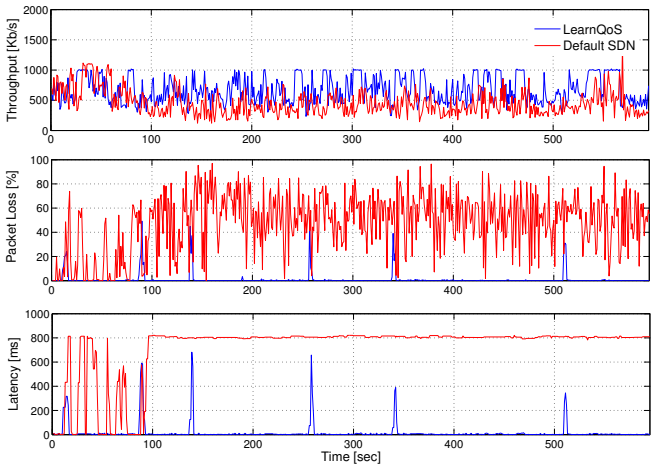


Fig. 3. Throughput, packet loss rate, and latency of QoS traffic flow for LearnQoS framework and default SDN without PBNM

and SSIM of the QoS video flow as well as the mapping to MOS done according to Table IV for both, LearnQoS and default SDN. The results indicate that when using the LearnQoS, the user perceives the video quality as *Good* based on both PSNR and SSIM to MOS mapping. Whereas in the case of default SDN, the user perceived quality for the QoS video flow is *Poor* (based on SSIM to MOS mapping) towards *Bad* (based on PSNR to MOS mapping). Thus, by using the proposed LearnQoS framework there is an increase of 161% in PSNR as compared to the default SDN.

Figure 4 illustrates a comparison snapshot of the QoS video frame from the original transmitted video, the video frame received using the LearnQoS framework and the video frame as received using the default SDN. It can be noticed that the QoS video frame quality becomes noticeably poorer relative to the original video frame when

TABLE V
AVERAGE PSNR TO MOS AND SSIM TO MOS MAPPING

| | Performance Metrics | | | |
|---|---|---|---|---|
| | **PSNR** | **MOS** | **SSIM** | **MOS** |
| **LearnQoS** | 41.92 | 4 (Good) | 0.98 | 4 (Good) |
| **Default SDN** | 16.03 | 1 (Bad) | 0.74 | 2 (Poor) |

the default SDN framework is used with a PSNR of 24.62dB indicating a *Poor* user perceived quality according to the MOS mapping in Table IV. However, by enabling the proposed LearnQoS framework the quality of the video frame improves considerably, with a PSNR of 49.73dB representing *Excellent* user perceived quality.

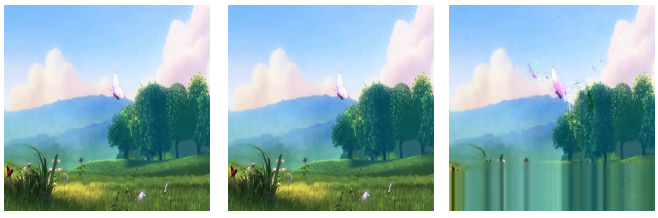


Fig. 4. Quantitative video frame quality comparison: a) original image, b) proposed LearnQoS framework (PSNR = 49.73dB, MOS = 5 - Excellent), and c) default SDN (PSNR = 24.62dB, MOS = 2 - Poor)

There is a trade-off between the monitoring overhead introduced and the application performance. For this purpose, we conducted several experimental runs using the same setup but with different monitoring update intervals, such as 1, 3, 6 and 9 seconds. The choice of the monitoring update interval values, starting from 3 seconds above is done because the SDN controller from the experimental setup needs time to perceive a consistent image of the entire network and to take the necessary measures to avoid the aftermath of policy violation. The results are listed in Table VI for the default SDN and the proposed LearnQoS framework. The results indicate that as the monitoring update interval increases the application performance decreases. This is because the SDN controller will take longer to detect and respond to the misbehaving best-effort traffic that affects the quality of the QoS video flow. However, even with the increased monitoring update interval LearnQoS outperforms the default SDN. For example, for a monitoring update interval of 6 seconds the quality of the QoS video flow is still perceived as *Good* when using LearnQoS, compared to *Bad* (based on PSNR to MOS mapping) towards *Poor* (based on SSIM to MOS mapping) as perceived when the default SDN framework is used.

Figure 5 shows the overall amount of monitoring overhead introduced on the control path for different monitoring update intervals. In particular, messages of type *OFPT_STATS_REQUEST* and *OFPT_STATS_REPLY* are used for measuring the throughput and packet loss, while the latency measurement is based on injecting

TABLE VI
AVERAGED PERFORMANCE EVALUATION FOR DIFFERENT MONITORING UPDATE INTERVALS (1, 3, 6, AND 9 SECONDS)

| **Performance Metrics** | Default SDN | LearnQoS Framework | | | |
|---|---|---|---|---|---|
| | | 1 | 3 | 6 | 9 |
| Throughput [Kb/s] | 454 | 654 | 641 | 632 | 634 |
| Packet Loss [%] | 47.28 | 1.02 | 1.75 | 2.31 | 5.74 |
| Latency [ms] | 719.36 | 17.13 | 20.63 | 25.23 | 28.54 |
| PSNR [dB] | 16.03 | 41.92 | 37.42 | 34.54 | 32.17 |
| MOS (PSNR) | 1 (Bad) | 4 (Good) | 4 (Good) | 4 (Good) | 3 (Fair) |
| SSIM | 0.74 | 0.98 | 0.98 | 0.98 | 0.94 |
| MOS (SSIM) | 2 (Poor) | 4 (Good) | 4 (Good) | 4 (Good) | 3 (Fair) |

*OFPT_PACKET_OUT* messages as probe packets into the network and waiting for receiving *OFPT_PACKET_IN* messages by the controller. The results show that the monitoring overhead is inversely proportional to the update interval. For example, the communication overhead is reduced by up to 89% when the update interval changes from 1 to 9 seconds. However, this comes at the cost of twice the packet loss rate and 23% decrease in PSNR. Thus, the trade-off between the introduced overhead and the application performance needs to be considered. Although the proposed LearnQoS framework adds more network overhead than the default SDN, the results show that the performance of the QoS application is significantly improved.

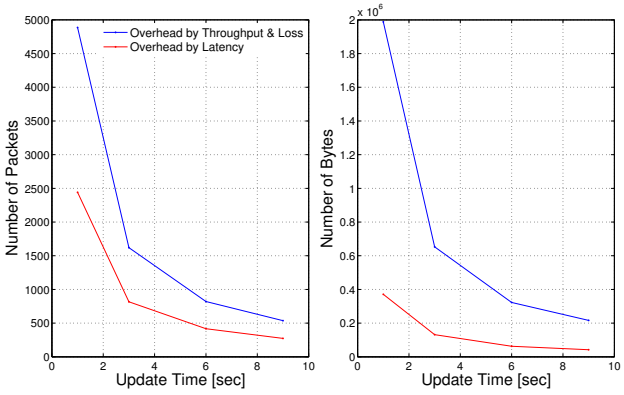


Fig. 5. Monitoring Overhead for different update intervals (1, 3, 6, and 9 seconds)

## VI. CONCLUSIONS

This paper proposes LearnQoS, a framework that makes use of RL to automate the management of PBNM over multimedia-based SDNs. LearnQoS was implemented and evaluated under an experimental setup based on Mininet, Floodlight controller and OpenvSwitch switches. For the performance evaluation, we compare the performance of the proposed LearnQoS framework against the default configuration of the SDN-based network in terms of PSNR, SSIM, MOS, throughput, packet loss and latency. The results show that the proposed LearnQoS framework outperforms the default multimedia-based SDN. Future works will consider a larger scale SDN-based network.

## REFERENCES


[1] R. Trestian, I. S. Comsa, and M. F. Tuysuz, "Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?" *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[3] I. S. Comsa, A. De-Domenico, and D. Ktenas, "Qos-driven scheduling in 5g radio access networks - a reinforcement learning approach," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[4] V. Martín, J. Cabrera, and N. García, "Design, optimization and evaluation of a q-learning HTTP adaptive streaming client," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 4, pp. 380–388, November 2016.

[5] M. Latah and L. Toker, "Application of artificial intelligence to software defined networking: A survey," *Indian Journal of Science and Technology*, vol. 9, no. 44, 2016.

[6] T. Uzakgider, C. Cetinkaya, and M. Sayit, "Learning-based approach for layered adaptive video streaming over sdn," *Computer Networks*, vol. 92, pp. 357–368, 2015.

[7] S. Sendra, A. Rego, J. Lloret, J. M. Jimenez, and O. Romero, "Including artificial intelligence in a routing protocol using software defined networks," in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 670–674.

[8] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "Qos-aware adaptive routing in multi-layer hierarchical software defined networks: a reinforcement learning approach," in *Services Computing (SCC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 25–33.

[9] A. Kucminski, A. Al-Jawad, P. Shah, and R. Trestian, "Qos-based routing over software defined networks," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–6.

[10] A. Al-Jawad, P. Shah, O. Gemikonakli, and R. Trestian, "Compression-based technique for sdn using sparse-representation dictionary," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 754–758.

[11] A. Al-Jawad, R. Trestian, P. Shah, and O. Gemikonakli, "Baprobsdn: A probabilistic-based qos routing mechanism for software defined networks," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–5.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[13] Mininet openflow virtual network. Accessed on November 1, 2017. [Online]. Available: http://mininet.org

[14] Floodlight openflow controller. Accessed on November 1, 2017. [Online]. Available: http://www.projectfloodlight.org/

[15] Open vSwitch. Accessed on April 1, 2018. [Online]. Available: http://openvswitch.org

[16] Ostinato traffic generator tool. Accessed on June 1, 2017. [Online]. Available: http://mininet.org

[17] C. V. N. Index, "Forecast and Methodology, 2015–2020 White Paper, Cisco, 2016," 2016.

[18] 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.

[19] 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.