# REDO: A Reinforcement Learning-based Dynamic Routing Algorithm Selection Method for SDN

Ahmed Al-Jawad[1], Ioan-Sorin Comșa[2], Purav Shah[1], Orhan Gemikonakli[1], Ramona Trestian[1]

[1]Middlesex University, London, UK
aa3512@live.mdx.ac.uk,{p.shah, o.gemikonakli, r.trestian}@mdx.ac.uk

[2] Swiss Distance University of Applied Sciences, Brig, Switzerland
ioan-sorin.comsa@ffhs.ch

***Abstract*—The current increase in the Internet traffic along with the global crisis have accelerated the roll-out of the next generation 5G network and key enabling technologies. In this context, addressing the end-to-end Quality of Service (QoS) provisioning in order to guarantee a sustainable service delivery to the end-users became of paramount importance. Some of the enabling technologies that could play a key role in this regard are Software Defined Network (SDN) and Machine Learning (ML). This paper proposes REDO, a Reinforcement lEarning-based Dynamic rOuting algorithm selection method that decides on the conventional routing algorithm to be applied on the traffic flows within a SDN environment. REDO will dynamically select the most appropriate routing algorithm from a set of centralized routing algorithms (MHA, WSP, SWP, MIRA) that maximizes the reward function from the network. The proposed REDO solution is implemented and evaluated using an experimental setup based on Mininet, Floodlight controller and Open vSwitch switches. The results show that REDO outperforms other state-of-the-art solutions.**

***Index Terms*—SDN, ML, QoS, Routing Algorithms**

## I. Introduction

Looking at the recent advancements in network technologies, it can be noticed that the classical network architecture has evolved towards a complex architecture with vendor-specific designed interfaces to accommodate the ever increasing traffic demands. According to Cisco, video traffic will reach 82% of all IP traffic by 2022 [1]. Thus, it becomes essential to consider Quality of Service (QoS) requirements within the network management functions. Unlike the previous generations, the fifth generation (5G) of mobile networks provides a new foundational architecture with stringent requirements of Network Function Virtualisation (NFV), massive scalability, high reliability and added flexibility [2]. Therefore, Software Defined Network (SDN), NFV and Artificial Intelligence/Machine Learning (AI/ML) have become key components in the design of next generation networks.

Previous works have investigated the use of Machine Learning for network routing [3], [4]. Uzakgider et al. [5] introduce a Reinforcement Learning (RL)-based routing algorithm that determines when to re-route the traffic to minimize the packet loss. Experimental results show that the proposed system achieves better results when compared to the shortest path routing and greedy-based approaches. However, complex scenarios with large-scale topologies are not addressed in this study. Similarly, Sendra et al. [6] propose an intelligent routing protocol for SDN based on RL. Whereas, Lin et al. [7] introduce a RL-based QoS-aware adaptive routing in a multi-layer hierarchical SDN environment. Hossain et al. [8] proposed a RL-driven QoS-aware routing algorithm to detect and prevent link congestion. The proposal is evaluated under normal and congestion scenarios and the results show that the proposed approach outperforms the Dijkstra algorithm-based method. Similarly, the work in [9] uses an AI prediction mechanism to determine the congestion expectation and also studies the path optimization for finding the route in the network to avoid congestion. Kumar et al. [10] explores the use of ML algorithms (i.e. K-means clustering and cosine similarity) to select the least congested route in SDN from a list of possible paths.

In contrast to the related works, this work proposes **REDO**, a **R**einforcement l**E**arning-based **D**ynamic r**O**uting algorithm selection method that makes use of the Q-learning as a RL algorithm. The proposed algorithm is trained to decide on the most suitable conventional routing algorithm to be applied on the traffic flows within a SDN environment. The inclusion of reinforcement learning in the SDN-based environment increases the cognitive abilities of the decision-making procedure [11]. The proposed approach is not focusing on designing a new routing algorithm that meets multiple constraints. Instead, REDO decides intelligently on the routing algorithm to be applied based on the reward that complies with the Service Level Agreement (SLA) requirement of service.

## II. Proposed REDO Framework

Figure 1 illustrates the framework of the proposed REDO solution built on top of the SDN architecture, consisting of: **(1) REDO** - the proposed **R**einforcement l**E**arning-based **D**ynamic r**O**uting algorithm selection block that makes use of Q-learning to decide on the most suitable routing algorithm to be applied in the network from a set of routing algorithms (i.e., Minimum Hop Algorithm (MHA) [12], Widest Shortest Path (WSP) [12], [13], Shortest Widest Path (SWP) [12], [13], and Minimum Interference Routing

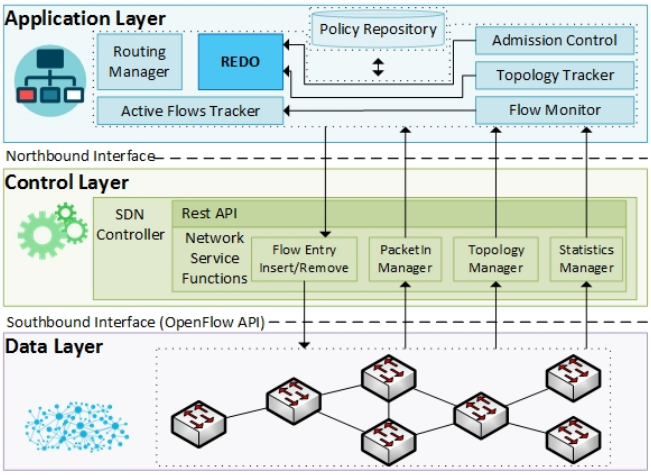


Fig. 1. Proposed SDN-based REDO Framework

Algorithm (MIRA) [14]); **(2) Routing Manager** - reroutes the active flows based on the output from REDO; **(3) Policy Repository** - stores the SLO (Service Level Objective) policy rules describing the technical interpretation in measurable terms (e.g. throughput, packet loss, etc.); **(4) Topology Tracker** - maintains a global image of the instantaneous network state and maps the physical network diagram to its graphical representation; **(5) Admission Control** - accepts/rejects incoming traffic requests; **(6) Flow Monitor** - periodically collecting statistics of all flows and maintains the flow state; **(7) Active Flow Tracker** - tracks active/inactive flows in the network.

## III. System Model

### A. Problem Formulation

The SDN network in the data plane is modeled by a graph $G(V,E)$ where $E$ is the set of links and $V$ is the set of nodes, with each node representing an SDN switch. A routing algorithm will find a feasible path $P$ described by a set of links $P = l_1, ..., l_n$ that connect the source and destination nodes. Each link $l \in E$ has a finite capacity $C_l$ while the remaining available bandwidth $BW_l$ of link $l$ is determined by $BW_l = C_l - \sum a_f$, where $a_f$ is the total bit rate of the passing flow $f$. Each traffic flow $f$ belongs to a set of flows $F = (F_{qos} \cup F_{bkg})$ where $F_{qos}$ and $F_{bkg}$ represent the sets of QoS and background flows, respectively. Each flow $f$ is further classified according to the network services of certain traffic class $v$ (e.g. video, HTTP, FTP), which is denoted by $f_v$. The main objective is to route the flows in a network to maximize the flows that satisfy the SLA requirement in terms of throughput, packet loss and rejection rate. However, the optimization problem is subject to constraints that needs to be satisfied to solve the problem, such that:

$$\text{maximize} \quad \sum_{f \in F} \sum_{p \in P} u_{p,f}(x_f \cdot y_f \cdot z_f) \tag{1a}$$

$$\text{subject to} \quad \sum_{f \in F} d_{l,f} \cdot a_f \le C_l, \quad \forall l \in E, \tag{1b}$$

$$\sum_{p \in P} u_{p,f} = 1, \quad \forall f \in F, \tag{1c}$$

$$\sum x_f = 0, \quad \forall f \in F, \tag{1d}$$

$$\sum y_f = 0, \quad \forall f \in F, \tag{1e}$$

$$\sum z_f = 0, \quad \forall f \in F, \tag{1f}$$

$$x_f \in \{0,1\}, \quad \forall f \in F, \tag{1g}$$

$$y_f \in \{0,1\}, \quad \forall f \in F, \tag{1h}$$

$$z_f \in \{0,1\}, \quad \forall f \in F, \tag{1i}$$

$$d_{l,f} \in \{0,1\}, \quad \forall f \in F, \forall l \in E, \tag{1j}$$

$$u_{p,f} \in \{0,1\}, \quad \forall f \in F, \forall p \in P \tag{1k}$$

where $d_{l,f}$ is a decision variable with value 1 if the flow $f$ is passing along link $l$ and 0 otherwise; $C_l$ represents the capacity of link $l$ and $a_f$ is the total bit rate of flow $f$. Constraint (1b) indicates that the sum of throughput of the flows routed over a link $l$ should not exceed its capacity $C_l$. Constraint (1c) indicates that a flow in the network shall be routed on one path only, where $u_{p,f} \in \{0,1\}$ is a decision variable that takes the value $u_{p,f} = 0$ if path $p$ is not selected by flow $f$, and the value $u_{p,f} = 1$, otherwise. Constraints ((1d))-((1f)) are defined to indicate that the active flow $f$ should satisfy the SLA requirement. Where $x_f$, $y_f$, and $z_f$ are decision variable with value 0 if flow $f$ satisfies the requirement $Q_{f,thr}$, $Q_{f,loss}$, and $Q_{f,rej}$ respectively and 1 otherwise. Constraints ((1g))-((1k)) indicate the variable based on a binary selection.

Solving the above problem using the RL approach brings several benefits compared to the traditional methods (e.g. heuristics). For example, RL algorithm is used for solving sequential decision problems without knowledge about the analytical model of the underlying system. Furthermore, RL is well designed for learning to optimize the problem [15] and the generalization by RL is much more flexible [16].

### B. RL-Based Solution

RL enables an agent to take an action in order to maximize a defined reward function. By this, the RL algorithm chooses and applies the action on the current state. As a result, the SDN system moves into the next state and the reward function evaluates the system performance and updates the value of the action selection in the previous state. In the following, the state space, action, and the reward function are defined.

*1) state space:* In Q-learning, the system states are mapped to actions in order to maximize the long term reward. Thus, the system state $S$ is defined as:

$$S = [\psi,\ \beta_{qos},\ \alpha_{qos},\ \phi_{qos}] \tag{2}$$

where $\psi \in \{low,\ medium,\ high\}$ is the traffic load. $\beta_{qos}$ indicates if the throughput requirement is met for QoS service type. Similarly, $\alpha_{qos}$ indicates if the packet loss rate requirement of QoS service type is met. Finally, $\phi_{qos}$ shows

if the rejection ratio is satisfying a certain level. The function of each parameter is given as follows:

$$\beta_{qos} = \begin{cases} 1 & \text{if } \sum x_{f_{qos}} = 0, \\ 0 & \text{if } \sum x_{f_{qos}} > 0 \end{cases} \tag{3}$$

$$\alpha_{qos} = \begin{cases} 1 & \text{if } \sum y_{f_{qos}} = 0, \\ 0 & \text{if } \sum y_{f_{qos}} > 0 \end{cases} \tag{4}$$

$$\phi_{qos} = \begin{cases} 1 & \text{if } \sum z_{f_{qos}} = 0, \\ 0 & \text{if } \sum z_{f_{qos}} > 0 \end{cases} \tag{5}$$

*2) action space:* The action space contains a set of routing algorithms $O_{qos}$. The action taken on the state at time $t$ can be denoted as $o_{qos}(t)$, where $o_{qos}(t) \in O_{qos}$ represents the routing algorithm applied on the QoS flow $f_{qos}$ at time $t$. The action is applied on the QoS traffic only, while for the rest of the traffic the routing strategy is static.

*3) reward function:* When an action is executed on a given state, the system shall observe in the upcoming time a new state of the network and it receives a reward as a feedback. The reward is determined by a function that maps the performance of an action taken in a given state into a scalar value by indicating how good the applied action is on that state. The reward consists of three sub-rewards obtained independently. The first sub-reward function describes how much the measured throughput of a flow varies from the SLA requirement and is defined as:

$$R_{TH,f_v} = \begin{cases} 1 - \left[ \frac{q_{v,thr} - \tilde{a}_{f_v}}{q_{v,thr}} \right] & if \quad \tilde{a}_{f_v} \le q_{v,thr} \\ 1 & if \quad \tilde{a}_{f_v} > q_{v,thr} \end{cases} \tag{6}$$

where $\tilde{a}_{f_v}$ is the measured throughput of flow $f_v$ and $q_{v,thr} \in Q_f$ is the minimum throughput requirement of a certain traffic class $v$. If the requirement of a flow is met, the reward function returns the highest possible reward value of 1.

Similarly, the second sub-reward represents the flow performance in terms of the packet loss rate defined as:

$$R_{PL,f_v} = \begin{cases} 1 - \left[ \frac{\tilde{b}_{f_v} - q_{v,loss}}{\tilde{b}_{f_v}} \right] & if \quad \tilde{b}_{f_v} \ge q_{v,loss} \\ 1 & if \quad \tilde{b}_{f_v} < q_{v,loss} \end{cases} \tag{7}$$

where $\tilde{b}_{f_v}$ is the measured packet loss rate of a flow $f_v$ that belongs to the traffic class $v$, while $q_{v,loss} \in Q_f$ is the maximum packet loss requirement. On the other hand, the third sub-reward is based on the rejection rate for a specific traffic class $v$ and is given by:

$$R_{RR,v} = \begin{cases} 1 - \left[ \frac{\tilde{c}_v - q_{v,rej}}{\tilde{c}_v} \right] & if \quad \tilde{c}_v \ge q_{v,rej} \\ 1 & if \quad \tilde{c}_v < q_{v,rej} \end{cases} \tag{8}$$

where $\tilde{c}_v$ is the measured rejection rate that belongs to the traffic class $v$, while $q_{v,rej} \in Q_f$ is the rejection rate requirement.

The overall reward for each traffic class $v$, is computed based on the following equation:

$$R_v = w_{TH} * \frac{\sum_{f_v \in F_v} R_{TH,f_v}}{N} + w_{PL} * \frac{\sum_{f_v \in F_v} R_{PL,f_v}}{N} + w_{RR} * R_{RR,v} \tag{9}$$

where $w_{Th}$, $w_{PL}$ and $w_{RR}$ represent the weights for the throughput, packet loss, and rejection rate respectively. Finally, the total reward is computed as a weighted sum of rewards of all traffic classes.

For the proof-of-concept in this work, four routing algorithms MHA, WSP, SWP and MIRA are defined in the action set $O_{qos} = \{MHA,\ WSP,\ SWP,\ MIRA\}$ for routing the QoS-based traffic. While the background traffic is routed using MIRA. The QoS service type is represented by the HD video traffic class, while the background service type is represented by SD video, HTTP, and FTP traffic classes. Thus, the traffic class $v \in \{HD\ video,\ SD\ video,\ HTTP,\ FTP\}$. The weights $w_{Th}$=$w_{PL}$=$w_{RR}$ = 1/3 are assumed to be equally important. While the total reward is given by:

$$R = \underbrace{w_{HD\_Video} * R_{HD\_Video}}_{QoS\ service\ type} + \underbrace{w_{SD\_Video} * R_{SD\_Video} + w_{ftp} * R_{FTP} + w_{http} * R_{HTTP}}_{Background\ service\ type} \tag{10}$$

where the traffic class weights are assigned based on the traffic ratios estimated by Cisco as detailed next.

In the training stage, the phase was executed on 60 individual trials. An individual trail is defined as a test scenario of a total run time of 1500 seconds. With respect to the traffic, the setup generates for each trail new values of the random seed in order to get a random set of traffic. The discount factor determines how much to weigh the value of maximum expected future rewards on the cumulative rewards. The discount factor is chosen near 1 to ensure convergence to the optimal policy. For the purpose of this study, the discount factor is set to $\lambda = 0.9$ in order to let the agent propagate long-term rewards. On the other side, the learning rate determines how fast the model learns from the changes imposed by the environment. In this study, the learning rate is set to $\alpha = 0.01$.

## IV. Experimental Setup

### A. Test Environment

The performance evaluation of the proposed REDO framework is done through an experimental setup consist-

ing of three main elements: (i) Mininet [1] - used to emulate the SDN data plane; (ii) external Floodlight OpenFlow controller [2] - provides RESTful API and network services; and (iii) the application layer - containing the network management for performance evaluation. The entire experiment is hosted on a powerful machine to accommodate the traffic load. The SDN controller and the entire application layer run on a virtual computer (2.2GHz multiprocessor of 4 CPU units with memory size of 16GB), while the Mininet test-bench is running on another virtual machine (2.2GHz multiprocessor of 4 CPU units with memory size of 32GB). Each virtual machine is running Linux-Ubuntu Server. Open vSwitch [3] is used as a software SDN switch.


Fig. 2. AT&T network topology used in the experimental setup

To emulate a real dynamic network environment, the AT&T topology from Internet Topology Zoo [17] was used as seen in Fig. 2. The network nodes are replaced by SDN-Openflow switches. Each switch has a host directly connected that generates data traffic. Two types of services are generated: guaranteed QoS-based service consisting of live HD video streaming and background services consisting of buffered SD video streaming, web browsing and file transfer traffic. In order to generate live HD and buffered SD video, VLC player tool is employed with a CBR encoder. The video source is created by using the FFMPEG video and audio converter [4]. On the other hand, HTTP and FTP traffic are generated using Ostinato [5] traffic generator tool. Based on the traffic classes, it is possible to evaluate different traffic mix and load on the network.

According to Cisco forecast, video traffic volume will reach 82% of all IP traffic by the year 2022 [1]. Based on these statistics, the traffic mix ratio in our experiment setup is determined such that 82% of the total traffic is represented by video traffic and the remaining 18% is represented by HTTP and FTP traffic. Additionally, the total volume of 82% video traffic can be divided into 63% live HD video and 19% buffered SD video [18]. The remaining of 18% is divided equally between HTTP and FTP traffic. The same ratios are maintained under different traffic loads. The parameters for live HD are as follows: 665Kbps average bit-rate, 24 frames per sec., and a resolution of 1280x720 pixels. The buffered SD video has an average bit-rate of 285Kbps, 24 frames per sec., and a resolution of 640x360 pixels. Both videos have a duration of 5 minutes. The parameters for the HTTP and FTP traffic model are chosen based on [19].

The performance of the proposed REDO solution is compared against the four routing algorithms MHA, WSP, SWP and MIRA in terms of throughput, packet loss, flow rejection, PSNR and Mean Opinion Score (MOS). A mapping of PSNR to MOS, used to subjectively assess the users' Quality of Experience (QoE) is given in Table I [20].

TABLE I
PSNR AND SSIM TO MOS MAPPING [20]

| 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) | $< 18.7$ | $< 0.5$ |

### B. Evaluation Scenarios

The total experiment duration is set to 1500 seconds. The destination node is chosen at random other than the source node within the network. In order to maintain the traffic mix ratio, each link in the topology operates at the speed of 1 Mb/s. A larger link capacity in the topology requires a higher number of HTTP and FTP flows to sustain the traffic ratio. In order to evaluate the routing algorithms under dynamic network conditions, three different levels of network load are considered such as: 0.5 (low load), 0.75 (medium load), and 1.0 (high load). The network load is computed based on the link load, link capacity and the number of links within the network topology.

The minimum throughput requirements for each traffic class is set to 658Kbps for live HD video, 279Kbps for buffered SD video, 14Kbps for Web browsing and 180Kbps for file transfer. In general, video is considered sensitive to network degradation. In order to satisfy the human perception, video quality becomes noticeable at packet loss of 0.5% and annoying when greater than 2% [21]. Based on this, the maximum acceptable packet loss rate is defined for live HD video traffic as 1% and 2% for the buffered SD video traffic. Other background traffic like HTTP and FTP have guarantees of zero packet loss rate. In terms of rejection rate requirement, the SLO policy is defined for the entire traffic and is set to 25% for the QoS video traffic and 35% for the background video traffic.

## V. RESULTS AND DISCUSSIONS

There are two phases involved in the RL process: training and testing. The training phase is used to learn the

[1]Mininet-http://mininet.org
[2]Floodlight-: http://www.projectfloodlight.org
[3]ovsswitch-http://openvswitch.org
[4]FFMPEG-tool," https://ffmpeg.org
[5]Ostinato-https://ostinato.org/

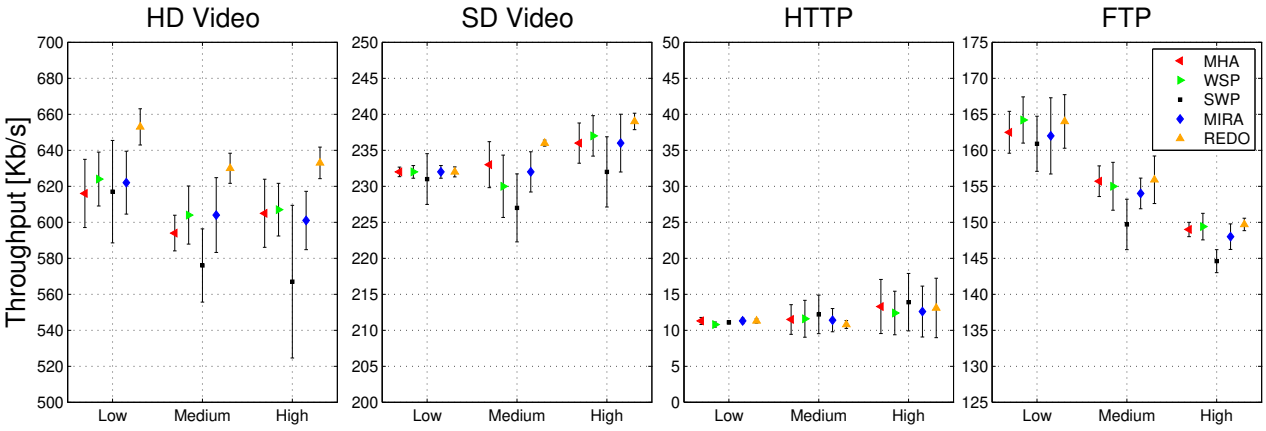


Fig. 3. Throughput of the traffic classes under different traffic loads

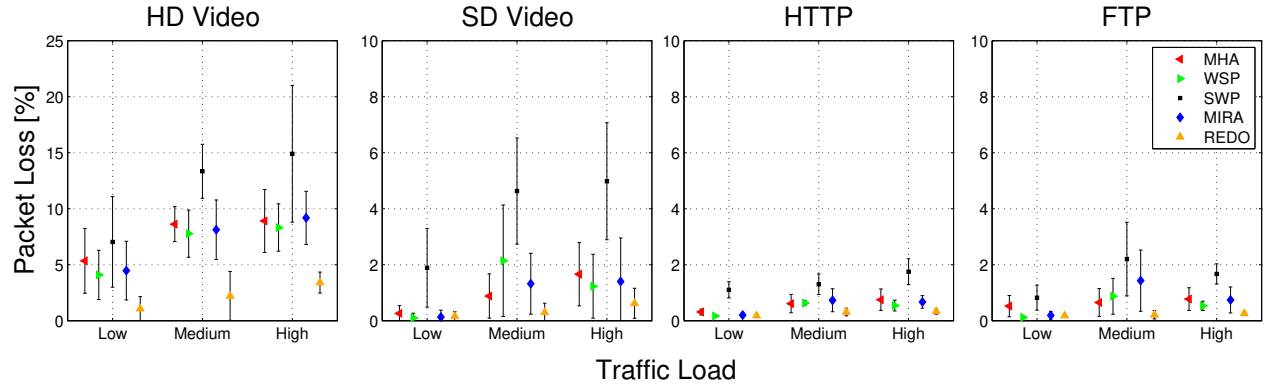


Fig. 4. Packet loss of the traffic classes under different traffic loads

TABLE II
AVERAGED ESTIMATED PSNR AND MOS UNDER DIFFERENT TRAFFIC LOADS, WHERE L = LOW LOAD, M = MEDIUM LOAD, AND H = HIGH LOAD

| | | MHA | | | WSP | | | SWP | | | MIRA | | | REDO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | l | m | h | l | m | h | l | m | h | l | m | h | l | m | h |
| HD | PSNR [dB] | 25.4 | 21.3 | 21 | 27.8 | 22.2 | 21.6 | 23 | 17.5 | 16.5 | 27 | 21.8 | 20.7 | 39.4 | 33.1 | 29.4 |
| | MOS | Poor | Poor | Poor | Fair | Poor | Poor | Poor | Bad | Bad | Poor | Poor | Poor | Good | Good | Fair |
| SD | PSNR [dB] | 51.7 | 41.1 | 35.6 | 60 | 33.4 | 38.2 | 34.5 | 26.7 | 26.1 | 57.7 | 37.6 | 37.1 | 56 | 50.5 | 44.2 |
| | MOS | Exc. | Good | Good | Exc. | Good | Good | Good | Poor | Poor | Exc. | Good | Good | Exc. | Exc. | Good |

optimal policy that maximizes the long-term reward. In the experimental test, the training phase was executed on 60 individual trials for each given scenario that is defined by traffic load. An individual trail is defined as a test scenario of a total run time of 1500 seconds. With respect to the traffic, the setup generates for each trail new values of the random seed in order to get a random set of traffic. In order to have a fair exploration of all possible state-action pairs, the $\epsilon$-greedy was set to zero. This means that all actions are randomly chosen in all system states. Once the system is trained, the testing phase is executed. In this phase, the algorithm exploits the learned Q-table based on the actual networking states. In this case, the $\epsilon$ value is 1. In order to compare fairly the five routing algorithms under various baseline factors (e.g. traffic load), the same sequence of experiment condition are ran on each scenario. The results were averaged over 5 simulation trails per scenario.

Figures 3 and 4 show that the proposed REDO solution outperforms the other routing algorithms in terms of throughput and packet loss. For example, under low load, REDO drastically reduces the packet loss to 1.07% for the QoS-based services while MHA, WSP, SWP, MIRA achieves an average packet loss of 5.34%, 4.09%, 7.03%, and 4.47%, respectively. As shown in Table II, this implies

accordingly an estimated averaged PSNR of 39.4dB for the proposed REDO method. Thus, REDO makes a significant improvement in terms of minimizing the packet loss when compared to the classical routing algorithms. Figure 5 shows that all solutions lead to more rejections in the incoming flows of QoS-based traffic class when the network load increases. Due to the increase in the total amount of the generated video traffic while the network capacity stays fixed, this implies an increase in the flow rejection rate of the QoS-based services.

Overall, the results indicate that the proposed REDO outperforms other classical routing algorithms in terms of maximizing throughput and minimizing the packet loss when the network load increases from low to high. REDO provides a *Good* (see Table II) user perceived quality under low and medium traffic loads, and a *Fair* user perceived QoE under high traffic load without penalizing the other traffic classes. In contrast, all the other routing algorithms provide a *Fair* (e.g., WSP and MIRA) and *Poor* (e.g., MHA and SWP) user perceived QoE under low traffic load which drops to *Poor* (e.g., MHA, WSP, and MIRA) and *Bad* (e.g., SWP) user perceived QoE under medium and high traffic loads. Consequently, in order to accommodate more QoS-based traffic flows, the classical routing algorithms will sacrifice the users' perceived quality for this traffic class as well as will penalize the performance of the other traffic classes.

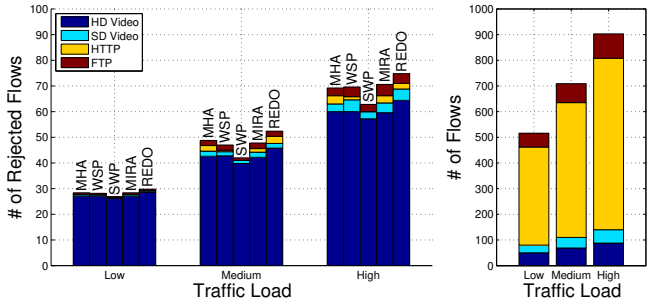


Fig. 5. The total number of rejected flow and the total number of flows that are generated in the experiment test.

## VI. Conclusions

This paper proposes **REDO**, a **R**einforcement l**E**arning-based **D**ynamic r**O**uting algorithm selection method that decides the most suitable routing algorithm to be applied on the traffic flows in SDN to enable QoS provisioning. REDO was implemented and evaluated using an experimental setup based on Mininet, Floodlight controller and Open vSwitch switches. Several scenarios are considered to demonstrate the benefits of REDO under realistic network conditions. When compared to other state-of-the-art routing algorithms (e.g., MHA, WSP, SWP, MIRA), the results show that on average REDO outperforms them and finds the best trade-off between throughput, packet loss and rejection rate for the QoS-based traffic class without penalizing the other background traffic classes.

## References


[1] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, p. 1, 2018.

[2] N. Al-Falahy and O. Y. Alani, "Technologies for 5g networks: Challenges and opportunities," *IT Professional*, vol. 19, no. 1, pp. 12–20, 2017.

[3] 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*, pp. 1–17, 2021.

[4] A. Al-Jawad, P. Shah, O. Gemikonakli, and R. Trestian, "Learnqos: A learning approach for optimizing qos over multimedia-based sdns," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2018, pp. 1–6.

[5] 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.

[6] 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.

[7] 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.

[8] M. B. Hossain and J. Wei, "Reinforcement learning-driven qos-aware intelligent routing for software-defined networks," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019, pp. 1–5.

[9] A. Guo, C. Yuan, G. He, and L. Xu, "Research on sdn/nfv network traffic management and optimization based on big data and artificial intelligence," in *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2018, pp. 377–382.

[10] S. Kumar, G. Bansal, and V. S. Shekhawat, "A machine learning approach for traffic flow provisioning in software defined networks," in *2020 International Conference on Information Networking (ICOIN)*. IEEE, 2020, pp. 602–607.

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[12] 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.

[13] M. Curado and E. Monteiro, "A survey of qos routing algorithms," in *Proceedings of the International Conference on Information Technology (ICIT 2004), Istanbul, Turkey*, 2004.

[14] M. Kodialam and T. Lakshman, "Minimum interference routing with applications to mpls traffic engineering," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 2. IEEE, 2000, pp. 884–893.

[15] K. Li and J. Malik, "Learning to optimize," *International Conference on Learning Representations (ICLR)*, 2017.

[16] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[17] 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.

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

[19] 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.

[20] 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.

[21] Troubleshooting packet loss: How much is an acceptable amount? Accessed on June 12, 2020. [Online]. Available: https://www.vyopta.com/blog/video-conferencing/understanding-packet-loss/