# An Innovative Reinforcement Learning-based Framework for Quality of Service Provisioning over Multimedia-based SDN Environments

Ahmed Al-Jawad, Ioan-Sorin Comşa, Purav Shah, Orhan Gemikonakli, Ramona Trestian

***Abstract*—Within the current global context, the coronavirus pandemic has led to an unprecedented surge in the Internet traffic, with most of the traffic represented by video. The improved wired and guided network infrastructure along with the emerging 5G networks enables the provisioning of increased bandwidth support while the virtualization introduced by the integration of Software Defined Networks (SDN) enables traffic management and remote orchestration of networking devices. However, the popularity and variety of multimedia-rich applications along with the increased number of users has led to an ever increasing pressure that these multimedia-rich content applications are placing on the underlying networks. Consequently, a simple increase in the system capacity will not be enough and an intelligent traffic management solution is required to enable the Quality of Service (QoS) provisioning. In this context, this paper proposes a Reinforcement Learning (RL)-based framework within a multimedia-based SDN environment, that decides on the most suitable routing algorithm to be applied on the QoS-based traffic flows to improve QoS provisioning. The proposed RL-based solution was implemented and evaluated using an experimental setup under a realistic SDN environment and compared against other state-of-the-art solutions from the literature in terms of throughput, packet loss, latency, peak signal-to-noise ratio (PSNR) and mean opinion score (MOS). The proposed RL-based framework finds the best trade-off between QoS vs. Quality of User Experience (QoE) when compared to other state-of-the-art approaches.**

***Index Terms*—Multimedia, QoE, QoS, Reinforcement Learning, Routing Algorithms, SDN**

## I. Introduction

The current global pandemic has unquestionably disrupted several industries with most countries implementing lockdown measures that forced their citizens to stay at home. These measures have changed the way Internet traffic is consumed [1], with millions of people staying at home and using the Internet for work, education, and entertainment. This has led to an increase in Internet traffic of up to 15-20% with a major growth in web conferencing, video, and gaming traffic classes [1].

This unprecedented increase in the Internet traffic along with the stringent Quality of Service (QoS) requirements of various application classes puts a tremendous pressure on the underlying networks making the provisioning of high performance Quality of Experience (QoE) to become one of the key challenges faced even by 5G networks. Due to the limitations in network resources and diverse range of applications, the different QoS requirements cannot be guaranteed and thus, have a great impact on the users' perceived QoE [2]. Consequently, guaranteeing QoS provisioning has become an active field of research especially for applications that require data delivery under certain QoS constraints (e.g., interactive multimedia, web conferencing, gaming, etc.). In this context, new emerging technologies and solutions are being explored to accommodate the high traffic demands, such as: Network Function Virtualisation (NFV) and Software Defined Networks (SDN) [3], [4], Advanced Television Systems Committee (ATSC) 3.0 [5], satellite back-haul [6], Multi-Access Edge Computing [7], Unmanned Aerial Vehicles (UAV) and drones [8], machine learning [9], [10].

SDN is seen as one of the key enabling technologies for 5G networks which brings several primary advantages, including: centralized network provisioning, network programmability via standardized interface, decoupling of control and forwarding planes, and feasibility through global network image [11]. The SDN-based environment represents a new area for network performance improvement, that lately has attracted both academia and industrial communities to investigate further challenges for an optimized network performance. On the other hand, the integration of machine learning (ML) and artificial intelligence (AI) has recently gained increasing popularity due to its applications in almost every sector [10], [12]–[14].

This paper introduces an innovative Reinforcement Learning (RL)-based framework for multimedia-based SDN environments that selects the most appropriate routing algorithm from a set of centralized routing algorithms that maximizes the return reward from the network and enables QoS provisioning. The proposed framework is implemented and evaluated against other state-of-the-art solutions from the literature, under a realistic environment in terms of throughput, packet loss, latency, rejection rate, peak signal-to-noise ratio (PSNR) and mean opinion score (MOS).

The rest of the paper is organized as follows. Section II presents the related work. In Section III, introduces the proposed RL-based system and Section IV, details the design of RL-based solution. Section V presents the experimental setup and evaluation scenarios while the results are discussed in Section VI. Finally, the conclusions are drawn

A. Al-Jawad, P. Shah, O. Gemikonakli, and R. Trestian are with Middlesex University, London, UK (e-mails: aa3512@live.mdx.ac.uk, {p.shah, o.gemikonakli, r.trestian}@mdx.ac.uk)

I.-S. Comşa is with the Institute for Research in Open-, Distance- and eLearning, Swiss Distance University of Applied Sciences, Brig, CH-3900, Switzerland (e-mail: ioan-sorin.comsa@ffhs.ch).

in Section VII.

## II. Related Work

Earlier studies have investigated the role of routing algorithms in traditional network architecture [15], [16]. However, with the evolution of SDN, many researchers have investigated the employment of SDN concept and addressed further its benefits. A performance analysis of SDN with the main focus on wireless networks is presented by Araniti et al. in [17]. The study in [18] presents a survey of different routing algorithms for dynamic settings of performance guaranteed traffic tunnels in backbone SDN networks. However, the performance evaluation is limited to specific network scenarios. Lee et al. [19] presented a comparative study of standard routing algorithms. However, the simulation setup is relatively simple. The work in [20] studied the performance analysis of SDN versus Open Shortest Path First (OSPF) networks. The OSPF networks are widely used for the intra-domain routing where the link-state information is advertised across the network routers to maintain a global view of the network topology. The results indicate better delay performance in large-scale networks when compared to SDN. One of the most common proposed method is the flow routing framework [21]–[30], where the flows' routes are adjusted dynamically according to the network state. Despite the available traffic engineering techniques like packet queuing, this solution uses the key features of SDN such as the global network view and the simplicity of network management. Egilmez et al. in [21]–[23] proposed OpenQoS, an end-to-end dynamic QoS routing solution for multimedia over SDN-based networks. This framework integrates route computation and management modules into the solution. As a result, the low prioritized packets are rerouted if the regular flows have impact on the video quality. Other works like [24]–[26] presented an approach for adaptive video streaming over SDN networks. The base and enhancement layers of a video flow are routed on different paths. When the video quality degrades, the base layer packets are rerouted on a feasible path that satisfies the QoS requirements while the enhancement layer packets stay on the same shortest path. Other related works like [31], [32] proposed a framework for QoS provisioning based on per-flow routing. Similarly the works in [27]–[30], [33]–[37] adopted the per-flow routing.

With the evolution of the SDN paradigm, the southbound and northbound interfaces are introduced for programming the underlined network platform. ML/AI applications can utilize this feature so that the intelligent-based network solutions can easily configure the lower-level network base. Additionally, the feature of maintaining a global view of the network has brought considerable benefits to the ML/AI applications that run on the SDN controller. Several studies in [38]–[44] present a survey on the application of AI techniques over the SDN paradigm to solve problems like load balancing and security. The studies show that the integration of AI techniques within SDN is promising, with several research groups introducing the use of reinforcement routing over SDN-based network.

Uzakgider et al. [45] introduce a routing algorithm based on RL that determines when to re-route the traffic to minimize the packet loss. The experimental results showed that the proposed system outperforms 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. [46] propose an intelligent routing protocol for SDN based on RL. Whereas, Lin et al. [47] introduce a RL-based QoS-aware adaptive routing in a multi-layer hierarchical SDN environment. Hossain et al. [48] proposed an RL-driven QoS-aware routing algorithm to detect and prevent link congestion. The proposal is evaluated under normal and congested scenarios and the results show that the proposed approach outperforms the Dijkstra-based method, which is widely used in the network routing applications due to its simplicity and efficiency. Dijkstra algorithm solves the single-source shortest path problem by finding the shortest path between the source and destination nodes in the network with non-negative edge weights [49]. Guo et al. [50] use an AI mechanism to predict the risk of congestion. Kumar et al. [51] explore a ML-based method for selecting the least congested route in SDN-based environments. Two methods based on K-means clustering and cosine similarity are utilized to select the least congested path from a list of possible paths. On the other hand, the work in [52] proposed a deep RL for routing optimization in SDN-based network. The approach utilizes the off-policy and actor-critic deep learning method. The path selection is taken according to the defined states represented by the traffic matrix based on the bandwidth request and the decision quality is evaluated based on the reward function that considers the average network delay.

In our previous work [53] we presented a comprehensive performance evaluation study of state-of-the-art routing algorithms over realistic multimedia-based SDN environments with dynamic network conditions and topology. The study showed that there is no one single routing algorithm that can perform best under all considered scenarios and networking conditions. Motivated by this, and in contrast to the current literature, this work proposes a new framework that integrates RL to provide a traffic management solution for end-to-end QoS provisioning. The approach is not focusing on designing a new routing algorithm that meets multiple constraints. Instead it utilizes the RL method to dynamically select the optimal routing algorithm from a set of routing algorithms, that achieves the best results under dynamic network conditions. It decides intelligently on the routing algorithm based on the reward that complies to the Service Level Agreement (SLA) requirements based on QoS parameters. The proposed RL-based framework satisfies the properties of autonomic system with self-configuration, self-healing and self-optimization [54], [55].

## III. System Architecture and Components

### A. Proposed RL-based Framework Architecture

Figure 1 illustrates the proposed RL-based framework built on top of the SDN architecture, that consists of:

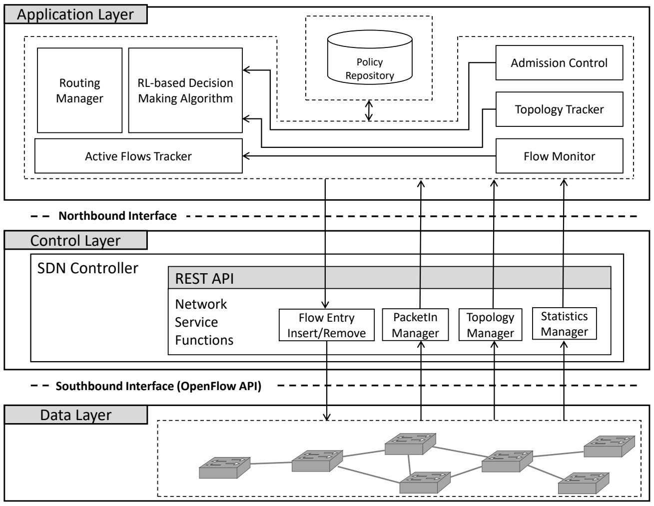



Fig. 1. Proposed RL-based framework under SDN architecture

**(1) RL-based Decision Making Algorithm** - makes use of Q-learning to add intelligence capability into the network. It decides on the most suitable routing algorithm to be applied from a set of routing algorithms. **(2) Routing Manager** - reroutes the active flows with the routing algorithm decided by the RL-based decision making algorithm. **(3) Policy Repository** - stores the Service Level Objective (SLO) policy rules that describe the technical interpretation in measurable terms (i.e. throughput, packet loss, rejection rate). **(4) Topology Tracker** - maps the physical network diagram to the graphical structural representation and it tracks a global image of the instantaneous network state. **(5) Admission Control** - responsible for accepting/rejecting incoming traffic requests. **(6) Flow Monitor** - maintains the flow state within the network by periodically collecting statistics of all flows. **(7) Active Flow Tracker** - tracks active/inactive flows in the network.

As proof of concept, four centralized routing algorithms are adopted and implemented into the framework, such as: (1) Minimum Hop Algorithm (MHA) chooses the path with the minimum number of links between the source and destination nodes [56]; (2) Shortest Widest Path (SWP) finds the feasible path with the maximum available bandwidth among the set of existing routes [56]. If there are multiple such paths, the one with the shortest route is selected; (3) Widest Shortest Path (WSP) finds the feasible path with the shortest path among the set of existing routes [56]. If there are multiple such paths, the one with the maximum available bandwidth is selected; (4) Minimum Interference Routing Algorithm (MIRA) exploits the knowledge of ingress egress pairs in order to minimize the interference between the paths when a new request arrives [57].

For the purpose of this work, the SLO requirements are defined directly without deriving them from the SLA. The translation and verification between SLA and SLO levels is out of the scope of this work. The framework maps the SLO policies to network policies by manipulating the flow tables of the SDN switches. The SLO policies are stored in an integrated database container.

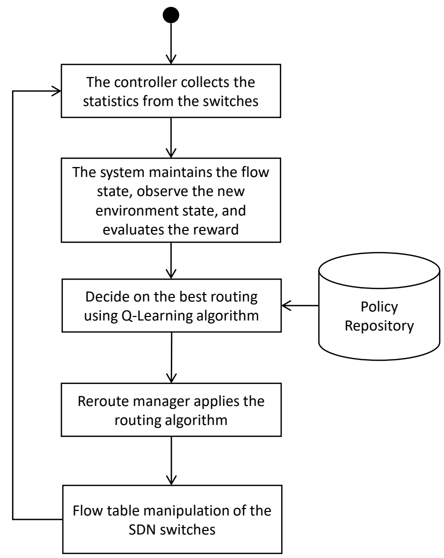



Fig. 2. Network management workflow

### *B. Network Management Function*

Two cases are identified for managing the network state: (1) *upon receiving a new route request* - The controller receives a packet-in message from a newly incoming request. The admission control in the application layer decides whether the request is accepted or rejected based on the resource availability. If the request is accepted, the traffic type is identified first and then the most suitable routing algorithm for the specific service type (e.g. QoS) is applied. (2) *upon monitoring the network state* - Figure 2 shows the work flow for this case. Initially, the flow monitor component periodically collects flow statistics from the network. The topology tracker builds a global view of the network state. Based on the current state, the Q-learning algorithm finds the best action (namely the routing algorithm) that returns the highest reward and it invokes the reroute manager to apply the current routing algorithm on the actual active flows in the network.

## IV. RL-Based Decision Making Solution

### *A. Problem Formulation*

Let the SDN data plane be modeled by an undirected graph $G(V,E)$, where $E$ is the set of links and $V$ is the set of nodes which represent the SDN switches. Each link in the network $l \in E$ is associated with a finite bandwidth capacity $C_l$, it indicates the maximum amount of flow that can pass through the link. Each traffic flow $f$ belongs to a set of flows $F = (F_{qos} \cup F_{bkg})$ where $F_{qos}$ stands for the QoS-based flows, while $F_{bkg}$ stands for the background flows. In general, a flow in the network is identified by 5-Tuple attributes (source and destination IP, source and destination port and the transport protocol) and it refers to data transmission between the source and destination nodes. However, each flow $f$ that belongs to a certain flow

TABLE I
NOTATION DEFINITION

| Variable | Definition |
|---|---|
| $G$ | The undirected graph |
| $E$ | The set of links |
| $V$ | The set of nodes |
| $F_{qos}, F_{bkg}$ | The set of QoS and background flows |
| $Q_{qos}, Q_{bkg}$ | The set of requirements for QoS and background service |
| $C_l$ | The bandwidth capacity of link $l$ |
| $a_f$ | The total bit rate or throughput of flow $f$ |
| $BW_l$ | The remaining available bandwidth of link $l$ |
| $P$ | The set of possible paths $p$ in the network |
| $l$ | The link in the network |
| $u_{p,f}$ | The path selection $p$ by the flow $f$ |
| $d_{l,f}$ | The link selection $l$ by the flow $f$ |
| $x_f$ | The throughput requirement is met by the flow $f$ |
| $y_f$ | The packet loss requirement is met by the flow $f$ |
| $z_f$ | The rejection rate requirement is met by the flow $f$ |
| $\gamma$ | topology size |
| $\psi$ | traffic load |
| $\beta_{qos}$ | The throughput requirement is met by all QoS flows |
| $\alpha_{qos}$ | The packet loss requirement is met by all QoS flows |
| $\phi_{qos}$ | The rejection ratio requirement is met by all QoS flows |
| $R_v, R$ | The total reward of traffic class $v$ |
| $R_{TH,f_v}$ | The sub-reward of throughput for traffic class $v$ |
| $R_{PL,f_v}$ | The sub-reward of packet loss for traffic class $v$ |
| $R_{RR,v}$ | The sub-reward of rejection ratio for traffic class $v$ |
| $O_{qos}$ | The set of actions applied on the QoS flows |
| $f_v$ | The traffic flow belongs to a certain traffic class $v$ |
| $v$ | The traffic class (e.g. video, HTTP, FTP) |
| $\tilde{a}_{f_v}$ | The measured end-to-end throughput of traffic class $v$ |
| $\tilde{b}_{f_v}$ | The measured packet loss rate of traffic class $v$ |
| $\tilde{c}_v$ | The measured rejection rate of traffic class $v$ |

set $\{F_{qos}, F_{bkg}\}$ is further classified according to the network services of certain traffic class $v$ (e.g. video, HTTP, FTP), this flow of a certain traffic flow is denoted by $f_v$. As data is transmitted through the network, 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 or the throughput of the passing flow $f \in F$. Table I shows the set of used variables in the problem definition.

The traffic flows in the network are requested by the network services or users in order to setup a feasible routing path. If $P$ is the set of possible paths, then the routing algorithm is used to find the feasible path $p \in P$, where a path is described by a set of links $p = l_1, ..., l_n$ that connects the source and destination nodes. Each flow in the network shall be routed on one path only, and therefore, the first constraints of our optimization problem are formulated as follows:

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

$$u_{p,f} \in \{0,1\}, \quad \forall f \in F, \forall p \in P, \tag{2}$$

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.

When the link becomes heavily loaded and congested due to multiple flows that are passing through the same link, then the involved traffic flows shall exhibit higher data loss and delay. To this extent, constraints (3) and (4) are introduced to indicate that the sum of throughput of all flows, passing through a given link $l$, should not exceed the maximum link capacity $C_l$.

$$\sum_{f \in F} d_{l,f} \cdot a_f \le C_l, \quad \forall l \in E, \tag{3}$$

$$d_{l,f} \in \{0,1\}, \quad \forall f \in F, \forall l \in E, \tag{4}$$

where $d_{l,f} \in \{0,1\}$ that indicates if a link $l$ is passed by a flow $f$ (i.e. $d_{l,f} = 1$ if the flow $f$ is passing along the link $l$ and $d_{l,f} = 0$, otherwise).

Network applications are usually associated to a set of service requirements $Q_f \in \{Q_{qos}, Q_{bkg}\}$, where requirement $q_f \in Q_f$ of flow $f$ is described for example by certain packet loss, delay or throughput level. Here $Q_{qos}$ stands for the requirement of QoS-based service type, while $Q_{bkg}$ stands for the requirement of background service type. The requirements vary based on the type of service and the tolerance level of user acceptance to the service, for example, multimedia applications can tolerate some amount of data loss, while financial application requires no data loss [58]. Here, constraints (5)-(10), and 10 are defined to indicate that the active flow $f$ should satisfy the SLA requirement. For this, we denote by $x_f \in \{0,1\}$ the decision variable set to $x_f = 0$ if flow $f$ of a certain traffic type satisfies the minimum throughput requirement $Q_{f,thr}$ and $x_f = 1$, otherwise. By $y_f \in \{0,1\}$, we define the decision variable with value $y_f = 0$ if $f$ satisfies the packet loss rate requirement $Q_{f,loss}$ and $y_f = 1$, otherwise. Finally, $z_f = \{0,1\}$ is a decision variable is set to a value of $z_f = 0$ if the rejection rate of the flow $f$ satisfies the requirement $Q_{f,rej}$, and $z_f = 1$, otherwise. Here, a given flow $f$ is assumed to belong to a certain set $\{F_{qos}, F_{bkg}\}$.

$$\sum x_f = 0, \quad \forall f \in F, \tag{5}$$

$$\sum y_f = 0, \quad \forall f \in F, \tag{6}$$

$$\sum z_f = 0, \quad \forall f \in F, \tag{7}$$

$$x_f \in \{0,1\}, \quad \forall f \in F, \tag{8}$$

$$y_f \in \{0,1\}, \quad \forall f \in F, \tag{9}$$

$$z_f \in \{0,1\}, \quad \forall f \in F \tag{10}$$

To further simplify the problem, this can be seen mainly as maximizing the utilization of all links in the network. Network utilization indicates how much of the network capacity is effectively utilized by active traffic flows. By this, the main parameter used to describe network utilization is the link utilization [59]. The overall optimization problem is formulated based on the constraints introduced previously:

$$\text{maximize} \sum_{l \in E} \sum_{f \in F} \frac{d_{l,f} \cdot a_f}{C_l}, \tag{11}$$

$$\text{subject to (1)-(10).} \tag{12}$$

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 the knowledge about the analytical model of the underlying system. Furthermore, RL is well designed for learning to optimize combinatorial problems [60]. Moreover, the generalization of decision-making given by RL is more flexible [61], [62].

### *B. RL-Based Solution*

RL as a type of machine learning technique is used to solve the sequential decision making problems. RL interacts with the dynamic environment and it improves iteratively its knowledge while exploring and observing the rewards and punishments from the environment. By this, it finds a suitable action model that would maximize the total cumulative reward of the agent [63]. In this research, RL is used to solve the optimization problem in (11) and (12) given the RL ability to deal with objective maximization problems [64]. Moreover, without having some specific rules to indicate the most appropriate routing algorithm each time, RL is considered as one of the best ML candidates to deal with such complex decision-making problems. Through RL, the best routing algorithm is learnt while interacting with the SDN environment based on the trial and error learning principle. Combining the optimization problem in (11) with the research problem defined in IV-A, the role of RL-based solution is to find the most suitable routing algorithm while the RL agent interacts with the network environment with the objective of maximizing the network utilization and respecting the QoS requirements for each flow.

To solve the optimization problem, the RL decision-making is achieved on a discrete state space, and thus, the state-action pairs can be enumerated exhaustively. Therefore, Q-learning as a model-free RL algorithm, is used to learn the most appropriate routing algorithm to be employed on each particular network state [64]. In decision-making problems with discrete state and action spaces, Q-learning converges to the optimal action selection on each state if all possible state-action pairs are visited for a consistent number of iterations [65]. Next, we introduce the state and action spaces, as well as the proposed reward function used to model the proposed decision-making problem.

*1) **State Space**:* Since the primary goal is to improve the QoS satisfaction of the active flows $f_{qos}$ with more stringent requirements $Q_{qos}$, then the system state $S$ is defined as:

$$S = [\gamma,\ \psi,\ \beta_{qos},\ \alpha_{qos},\ \phi_{qos}], \tag{13}$$

where $\gamma \in \{scale_{small},\ scale_{medium},\ scale_{large}\}$ is the topology size, $\psi \in \{load_{low},\ load_{medium},\ load_{high}\}$ is the size of traffic load. Parameter $\beta_{qos}$ indicates if the throughput requirement is met for the particular QoS service type. Similarly, the state parameter $\alpha_{qos}$ 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. All these parameters have a binary representation calculated as follows:

$$\beta_{qos} = \begin{cases} 1 & \text{if } \sum x_{f_{qos}} = 0, \\ 0 & \text{if } \sum x_{f_{qos}} > 0, \end{cases} \tag{14}$$

$$\alpha_{qos} = \begin{cases} 1 & \text{if } \sum y_{f_{qos}} = 0, \\ 0 & \text{if } \sum y_{f_{qos}} > 0, \end{cases} \tag{15}$$

$$\phi_{qos} = \begin{cases} 1 & \text{if } \sum z_{f_{qos}} = 0, \\ 0 & \text{if } \sum z_{f_{qos}} > 0. \end{cases} \tag{16}$$

*2) **Action Space**:* Since the objective is to find the best fitting routing algorithm that drives a long-term optimal solution for QoS flows, then the action space $O_{qos}$ is defined as a set of routing algorithms. As a proof of concept, four routing algorithms MHA, WSP, SWP and MIRA are considered in this paper, such as: $O_{qos} = \{MHA,\ WSP,\ SWP,\ MIRA\}$. The action taken on state $s_t$ at time $t$ is denoted as $o_{qos}(t)$, where $o_{qos}(t) \in O_{qos}$ stands for the routing algorithm applied on the QoS flow $f_{qos}$ at time $t$. The goal is to find the best action $o_{qos}(t) \in O_{qos}$ for the QoS service class such that, the overall QoS revenue in all service classes is maximized.

*3) **Reward Function**:* When an action is executed on a given state, the system shall observe a new state of the network and it receives a reward as a feedback. The reward is determined by a function that maps the action taken in a given state into a scalar value. More precisely, it measures the performance of the applied routing algorithm in a particular state. In this work, the proposed reward function is decomposed into three sub-rewards that are computed independently. The first sub-reward function measures the level of throughput reported to its associated SLA requirement, such as:

$$R_{TH,f_v} = \begin{cases} 1 - \left[\frac{q_{v,thr} - \tilde{a}_{f_v}}{q_{v,thr}}\right] & if \quad \tilde{a}_{f_v} \leq q_{v,thr} \\ 1 & if \quad \tilde{a}_{f_v} > q_{v,thr} \end{cases} \tag{17}$$

where $\tilde{a}_{f_v}$ is the measured throughput of flow $f_v$ that belongs to a certain traffic class $v \in \{HD\ video,\ SD\ video,\ HTTP,\ FTP\}$ and $q_{v,thr} \in Q_f$ is the minimum throughput requirement of a certain traffic class $v$. Here, for proof of concept, 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. If the requirement of a flow is met, the reward function returns the highest reward value of 1.

Similarly, the second sub-reward represents the flow performance in terms of the packet loss rate. The sub-reward is computed as follows:

$$R_{PL,f_v} = \begin{cases} 1 - \left[\frac{\tilde{b}_{f_v} - q_{v,loss}}{\tilde{b}_{f_v}}\right] & if \quad \tilde{b}_{f_v} \geq q_{v,loss} \\ 1 & if \quad \tilde{b}_{f_v} < q_{v,loss} \end{cases} \tag{18}$$

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 given by:

$$R_{RR,v} = \begin{cases} 1 - \left[\frac{\tilde{c}_v - q_{v,rej}}{\tilde{c}_v}\right] & if \quad \tilde{c}_v \geq q_{v,rej} \\ \\ 1 & if \quad \tilde{c}_v < q_{v,rej} \end{cases} \tag{19}$$

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:

$$R_v = w_{TH} * \frac{\sum_{f_v \in F_v} R_{TH,f_v}}{N} + w_{PL} * \frac{\sum_{f_v \in F_v} R_{PL,f_v}}{N} + w_{RR} * R_{RR,v} \tag{20}$$

where $w_{Th}$, $w_{PL}$ and $w_{RR}$ represent the weights of sub-rewards calculated for throughput, packet loss, and rejection rate, respectively. In this work it is assumed that all three parameters are equally important, and consequently, $w_{Th}$=$w_{PL}$=$w_{RR}$ = 1/3. Finally, the overall reward function is computed as the sum of rewards of all traffic classes $\{HD\ Video,\ SD\ video,\ HTTP,\ FTP\}$ as given by:

$$\begin{aligned} R = &\underbrace{w_{HD\_Video} * R_{HD\_Video}}_{QoS\ service\ type} \\ &+ \underbrace{w_{SD\_Video} * R_{SD\_Video} + w_{ftp} * R_{FTP} + w_{http} * R_{HTTP}}_{Background\ service\ type} \end{aligned} \tag{21}$$

The weights are assigned based on the traffic ratios in the setup. The ratios are provided by Cisco [66] as described later in Section V. For the QoS service represented by HD video traffic, the weight $w_{HD\_Video}$ is assigned to a ratio of 63%. For the background traffic, $w_{SD\_Video}$ is the weight for SD video assigned to the traffic ratio of 19%, while $w_{ftp}$ and $w_{http}$ are the weights for the web browsing and FTP traffic, respectively. Each weight is assigned to the traffic ratio of 9%.

## V. Experimental Setup and Scenarios

### A. Experimental Environment Setup

The experimental setup is used to evaluate the RL-based method (as described in IV). The overall test-bed consists of three main elements: (i) Mininet [67] - used to emulate the SDN data plane; (ii) external Floodlight OpenFlow controller [68] - provides RESTful API and network services like the flow entry update; and (iii) the application layer - containing the network management for performance evaluation. During the test execution, the relevant data is collected and stored for post processing and performance comparison. In this work, the entire experiment is hosted on OpenStack 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 on Linux-Ubuntu Server. Open vSwitch [69] is used as a software SDN switch. The Open vSwitch is implemented in kernel space with Linux. The switch is employed in the work to avoid the system call overheads between the user space and kernel space in Linux environment. The switch is commonly used with Mininet emulator.

The performance evaluation of the proposed method is performed under dynamic network conditions and over three realistic network topologies selected from Internet Topology Zoo [70] as illustrated in our previous work [53]:

- AT&T (large-scale topology): 25 nodes and 56 links;
- Sprint (middle-scale topology): 11 nodes and 18 links;
- GetNet (small-scale topology): 7 nodes and 8 links.

For each topology, the network nodes being replaced by SDN-Openflow enabled switches. Each switch has a host directly connected that generates data traffic.

### B. Traffic Characteristics

Two types of services are generated: QoS-based multimedia services and the background services. Under the two service types, four types of traffic are emulated: live HD video streaming as part of the QoS-based multimedia services and buffered SD video streaming, web browsing and file transfer traffic as part of the background services. In order to stream the live HD and buffered SD video streaming, VLC player tool is employed. The video streaming is represented by one-way transmission with a CBR encoder. The video source is created by using the FFMPEG video and audio converter [71]. While, HTTP and FTP traffic are generated using Ostinato [72] traffic generator tool and modelled according to [73], [74] and [73], respectively.

According to Cisco forecast, video traffic volume will reach 82% of all IP traffic by 2022 [75]. 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, based on the Cisco predictions in [66], this work assumes that the total volume of 82% video traffic is divided into 63% live HD video and 19% buffered SD video. The same ratios are maintained for different topologies and under different traffic loads. The parameters for live HD and buffered SD video traffic are listed in Table II. Due to the processing capacity limitations of the experimental setup and in order to maintain the traffic mix ratio based on the statistics provided by Cisco [75], each link in the topology operates at the speed of 1 Mb/s. This does not affect the evaluation performance and the approach can be scaled up to a larger network capacity, which will also require a higher number of traffic flows to sustain the defined traffic ratio. Consequently, the average bit-rate of the HD and SD video traffic have been scaled down to fit the network capacity of the experimental setup, while maintaining the ratio between them.

TABLE II
PARAMETERS OF VIDEO TRAFFIC

| Traffic | Parameters | Value |
|---|---|---|
| Live HD video | Average bit-rate [Kb/s] | 665 Kb/s |
| | Frame rate [fps] | 24 |
| | Resolution [pixels] | $1280 \times 720$ |
| | Duration [minutes] | 5 |
| Buffered SD video | Average bit-rate [Kb/s] | 285 Kb/s |
| | Frame rate [fps] | 24 |
| | Resolution [pixels] | $640 \times 360$ |
| | Duration [minutes] | 5 |

### C. Learning Stage

One of the main objective of RL is to train the agent from their experiences by interacting with their environment and improving its knowledge through trial and error [76]. In this work, the Q-learning method is employed as an RL technique in order to find the optimal action-selection policy that maximizes the discounted cumulative reward over time. For this, two phases are typically involved in the RL process: training and exploitation or testing. The training phase is used to learn the algorithm and find the optimal policy that maximizes the long-term reward. In the training phase, a large training data set is employed to learn the algorithm. While in the exploitation phase, the agent exploits the learned Q-table to choose the best action.

In general RL has a trade-off between exploration and exploitation. The 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 order to apply a fair exploration-exploitation trade-off, the $\epsilon$-greedy was set to zero in the training phase in order to explore more the environment. After the system is trained, the exploitation phase is executed afterwards. In this phase, the algorithm exploits the learned Q-table based on the actual network state. For the $\epsilon$-greedy, the $\epsilon$ value was set to 1 in the exploitation phase.

In the training stage, the phase was executed on 60 individual trials for each given scenario that is defined by certain topology size and traffic load (e.g., GetNet topology with low traffic load). By this, 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. According to the notations introduced in Subsection IV.B, the Q-learning algorithm updates the Q-table based on the following equation [76]:

$$\begin{aligned} Q[s_t, o_{qos}(t)] &= Q[s_t, o_{qos}(t)] \\ &+ \alpha \left\{ R + \lambda max_{o_{qos}} Q(s_{t+1}, o_{qos}) - Q[s_t, o_{qos}(t)] \right\} \end{aligned} \tag{22}$$

where $Q[s_t, o_{qos}(t)]$ represents the Q value of the state-action $(s_t, o_{qos}(t))$ pair. Let $s_t$ and $o_{qos}(t)$ denote the state and the action, respectively, executed by an agent at a time instant $t$. The reward earned from the environment is represented by $R$, while $max_{o_{qos}} Q(s_{t+1}, o_{qos})$ is the maximum estimated future reward given the next state $s_{t+1}$ and its all possible actions $o_{qos} \in O_{qos}$. At last, $\lambda$ and $\alpha$ represent the discount factor and the learning rate respectively, with values between 0 and 1.

The discount factor determines how much to weigh the value of maximum expected future rewards on the cumulative rewards. A discount factor closer to 0 results in higher preference to the immediate reward. By this, learning becomes weak and only the current knowledge is utilized in the decision making. While a discount factor equal to 1 makes the agent to consider all of its future rewards. In particular, the discount factor is chosen near 1 to ensure convergence to the optimal policy. For the study, the discount factor is set to $\lambda = 0.9$ in order to let the agent propagate long-term rewards [76].

On the other side, the learning rate determines how fast the model learns from the changes imposed by the environment. The learning rate of 0 means that the Q-values are never updated with the new reward, meaning that, the learning is not taking place. While a high value of the learning rate leads to the learning happening very quickly and the results become fluctuating and error-prone. In this study, the learning rate is set to $\alpha = 0.01$.

### D. Evaluation Scenarios

The performance of the proposed RL-based framework is compared against the individual performance of each of the four routing algorithms (e.g., MHA, WSP, SWP, MIRA) under realistic and dynamic network conditions. Several scenarios with different topologies and a mix of QoS and background flows are considered. Each scenario has a total experiment duration of 1500 seconds. The destination node is chosen at random other than the source node within the network. In the performance evaluation, the following parameters are considered in order to drive a dynamic network evaluation:

- **Network topology**: different network topologies are employed: AT&T (large-scale topology), Sprint (middle-scale topology), and GetNet (small-scale topology). The network topologies were taken from Internet zoo topology [70].
- **Service type**: Two service types are generated (QoS and background traffic) with four traffic classes associated to the service type. For the QoS-based traffic, live HD video streaming is employed. While background traffic is represented by buffered SD video streaming, web browsing and file transfer. For simplicity, traffic classification in the framework is based on the port identification to associate the incoming flow with the correct service type.
- **Network load level**: the number of active flows in the network is adjusted at any given moment in order to achieve the requested network load. The average load per one link is computed by dividing the current traffic load to the link capacity, while the total network load is calculated based on the average link load of the overall network. Three different configurations for the network load are

TABLE III
PSNR and SSIM to MOS Mapping [80]

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

considered: 0.5 (low load), 0.75 (medium load), and 1.0 (high load). The network load $NL$ is calculated as follows:

$$NL = \frac{\sum_i^N \frac{LL_i}{LC_i}}{N} \tag{23}$$

where $LL$ is the load over the link, $LC$ is the link capacity, and $N$ is the number of links in the network topology.

### E. Performance Metrics

The performance of proposed RL-based framework is assessed within a realistic multimedia-based SDN environment with dynamic network conditions such as different topology, traffic patterns and traffic load. The performance evaluation is done in terms of throughput [77], packet loss [78], delay [77], flow rejection rate [18], and PSNR [79]. A mapping of PSNR and SSIM to the nominal Mean Opinion Score (MOS) is given in Table III [80]. MOS is a five point scale used to subjectively assess the users' QoE [81].

The two services types considered in this work have different demands in terms of QoS requirements [82]. Generally, for the background traffic, the network does not provide any guarantee to deliver the packets to the destination [83]. Therefore QoS requirements based on the traffic service and class can be initially separated.

In general, video is considered sensitive to network degradation. To satisfy the human perception, video quality becomes noticeable at packet loss of 0.5% and annoying when greater than 2% [84]–[89]. Similarly, other studies in [90]–[92] indicate that video conferencing traffic with a packet loss between 1% and 2.5% is considered acceptable and above 4-6% packet loss it becomes irritating for the end user. Based on this, the maximum acceptable packet loss rate for QoS service class represented by live HD video traffic is set to 1%. On the other hand, in order to meet an acceptable video perception, the maximum acceptable packet loss for the buffered SD video traffic part of the background service class is set to 2%. Other background traffic like HTTP and FTP shall have a guarantees of zero packet loss rate [85], [88]. The minimum throughput requirement is obtained based on the average video bit-rate and the corresponding packet loss requirement. It is calculated based on:

$$q_{v,thr} = (100\% - q_{v,loss}) \times br_v \tag{24}$$

where $q_{v,loss}$ is the maximum packet loss requirement and $br_v$ is the total average video bit-rate, both associated to traffic class $v$. Table IV illustrates the set of requirements for

TABLE IV
Requirement set for QoS and background traffic

| Traffic Class | $q_{qos,thr}$ | $q_{qos,loss}$ | $q_{qos,rej}$ |
|---|---|---|---|
| QoS Service | | | |
| Live HD video | 658 Kb/s | 1% | 25% |
| **Traffic Class** | $q_{bkg,thr}$ | $q_{bkg,loss}$ | $q_{bkg,rej}$ |
| Background Service | | | |
| Buffered SD video | 279 Kb/s | 2% | 35% |
| Web browsing | 14 Kb/s | 0% | 35% |
| File transfer | 180 Kb/s | 0% | 35% |

QoS and background services. The rejection rate indicates the maximum acceptable rejection rate for that particular traffic class. The values were chosen to represent a reasonable rejection distribution among different traffic classes.

## VI. Results and Discussions

The proposed RL-based framework as described in section IV-B operates as follows: after the current network state is measured, the RL-based framework finds the optimal action for rerouting the QoS traffic under the actual network state while the background traffic is routed using MIRA for the entire duration of all the experiments.

The performance evaluation of the proposed RL-based framework was compared against the other four state-of-the-art routing algorithms (e.g., MHA, WSP, SWP, MIRA) under varying traffic load and topology networks in terms of average throughput, average packet loss, average PSNR. The average PSNR was estimated based on [93]. In the outcome results, the total number of flows indicates the total amount of generated flow under the trail for a certain service type, while the number of rejected flows indicates the number of flows rejected and no setup is carried on the network during the experiment trail.

As illustrated in the framework architecture in Subsection III-A, the flow monitor maintains the state of the network by periodically collecting the statistics of all flows in the network switches. In GetNet topology, the monitoring update interval of the flow monitor was set to 15 seconds. Due to the amount of traffic volume and the way to iterate through all switches to collect the flow statistics. Through several experimental runs, it has been observed that this value of 15 seconds is suitable to maintain a full image of the network state. Other lower values of the monitoring update interval would lead to incompleteness of dataset which leads to inaccurate results. Similarly, in order to monitor the network periodically in Sprint topology, the monitoring update interval was set to 15 seconds. While due to the topology size, high volume of traffic flows and processing limitation, the monitoring update interval was set to 45 seconds in AT&T topology.

In order to compare fairly the routing algorithms under various baseline factors (e.g. traffic load and network topology), 5 simulation trials for each single scenario (e.g.

TABLE V
THE MEAN AND STANDARD DEVIATION OF THE THROUGHPUT MEASUREMENT FOR THE ROUTING ALGORITHMS UNDER GETNET, SPRINT AND AT&T, WHERE L = LOW LOAD, M = MEDIUM LOAD, AND H = HIGH LOAD

| | MHA Throughput [Kb/s] | | | WSP Throughput [Kb/s] | | | SWP Throughput [Kb/s] | | | MIRA Throughput [Kb/s] | | | RL-based Method Throughput [Kb/s] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | l | m | h | l | m | h | l | m | h | l | m | h | l | m | h |
| **GetNet** | | | | | | | | | | | | | | | |
| HD | 650 | 651 | 624 | 650 | 651 | 626 | 650 | 649 | 625 | 650 | 651 | 626 | 650 | 651 | 631 |
| | ±3.96 | ±3.96 | ±10.2 | ±4.16 | ±1.17 | ±7.95 | ±3.62 | ±4.55 | ±8.79 | ±4.12 | ±2.17 | ±7.82 | ±4.11 | ±3.56 | ±12.57 |
| SD | 238 | 235 | 232 | 239 | 235 | 233 | 238 | 234 | 232 | 239 | 233 | 233 | 238 | 235 | 233 |
| | ±1.81 | ±0.61 | ±1.57 | ±1.18 | ±0.78 | ±1.71 | ±1.79 | ±1.58 | ±1.95 | ±1.76 | ±3.25 | ±1.32 | ±1.1 | ±0.87 | ±0.45 |
| HTTP | 15.8 | 15.3 | 16.2 | 16.4 | 15.3 | 16.3 | 15.9 | 15.6 | 16.1 | 16 | 15.7 | 16.3 | 15.8 | 15.5 | 15.9 |
| | ±1.5 | ±0.98 | ±0.5 | ±1.5 | ±1.39 | ±0.55 | ±1.73 | ±0.95 | ±0.39 | ±1.73 | ±3.25 | ±0.78 | ±1.79 | ±1.24 | ±0.6 |
| FTP | 172.9 | 185.6 | 178.4 | 173.6 | 184.4 | 177.3 | 172.2 | 184.4 | 178.8 | 173 | 184 | 179.1 | 172 | 184.4 | 178.6 |
| | ±3.95 | ±2 | ±3.46 | ±4.55 | ±1.85 | ±4.6 | ±3.67 | ±2.49 | ±3.15 | ±4.15 | ±2.04 | ±5.25 | ±3.6 | ±3.24 | ±4.55 |
| **Sprint** | | | | | | | | | | | | | | | |
| HD | 648 | 620 | 596 | 648 | 620 | 606 | 649 | 624 | 606 | 648 | 626 | 606 | 646 | 643 | 638 |
| | ±1.17 | ±11.0 | ±7.4 | ±1.18 | ±10.1 | ±8.6 | ±0.76 | ±11.6 | ±1.8 | ±1.52 | ±11.9 | ±13.4 | ±1.52 | ±2.0 | ±5.4 |
| SD | 236 | 235 | 233 | 236 | 235 | 233 | 236 | 234 | 234 | 236 | 235 | 234 | 236 | 234 | 235 |
| | ±0.51 | ±1.58 | ±1.3 | ±0.53 | ±0.93 | ±2.2 | ±0.16 | ±1.3 | ±1.5 | ±0.6 | ±1.3 | ±2.5 | ±0.51 | ±1.15 | ±1.77 |
| HTTP | 15.2 | 16 | 14.6 | 15.2 | 15.7 | 14.4 | 15.3 | 16.1 | 14.6 | 15.2 | 15.7 | 14.5 | 15.3 | 15.9 | 14.3 |
| | ±1.6 | ±1.0 | ±1.5 | ±1.25 | ±0.84 | ±1.9 | ±1.27 | ±0.84 | ±1.85 | ±1.42 | ±1.0 | ±2.0 | ±1.37 | ±1.0 | ±1.7 |
| FTP | 186.7 | 177.8 | 161.7 | 185.7 | 177.8 | 165.7 | 186.8 | 174.8 | 164 | 186.6 | 178.7 | 166.3 | 186.3 | 177.8 | 167.9 |
| | ±3.5 | ±7.6 | ±4.9 | ±2.6 | ±5.9 | ±5.0 | ±3.2 | ±5.2 | ±3.1 | ±2.72 | ±7.3 | ±4.7 | ±2.5 | ±7.6 | ±2.7 |
| **AT&T** | | | | | | | | | | | | | | | |
| HD | 616 | 594 | 605 | 624 | 604 | 607 | 617 | 576 | 567 | 622 | 604 | 601 | 653 | 630 | 633 |
| | ±18.9 | ±9.9 | ±18.9 | ±14.9 | ±16.1 | ±14.6 | ±28.4 | ±20.3 | ±42.4 | ±17.5 | ±20.8 | ±16.2 | ±10.0 | ±8.4 | ±8.7 |
| SD | 232 | 233 | 236 | 232 | 230 | 237 | 231 | 227 | 232 | 232 | 232 | 236 | 232 | 236 | 239 |
| | ±0.66 | ±3.2 | ±2.8 | ±0.87 | ±4.3 | ±2.8 | ±3.5 | ±4.72 | ±4.87 | ±0.88 | ±2.8 | ±4.0 | ±0.7 | ±0.42 | ±1.1 |
| HTTP | 11.3 | 11.5 | 13.3 | 10.8 | 11.6 | 12.4 | 11.1 | 12.2 | 13.9 | 11.3 | 11.4 | 12.6 | 11.3 | 10.8 | 13.1 |
| | ±0.5 | ±2.1 | ±3.8 | ±0.35 | ±2.6 | ±3.0 | ±0.23 | ±2.7 | ±4.0 | ±0.38 | ±1.6 | ±3.5 | ±0.3 | ±0.54 | ±4.1 |
| FTP | 162.5 | 155.7 | 149 | 164.2 | 155 | 149.4 | 160.9 | 149.7 | 144.6 | 162 | 154 | 148 | 164 | 155.9 | 149.7 |
| | ±2.9 | ±2.1 | ±1.0 | ±3.2 | ±3.3 | ±1.85 | ±3.8 | ±3.5 | ±1.6 | ±5.3 | ±2.1 | ±1.8 | ±3.72 | ±3.3 | ±0.866 |

MHA routing algorithm under GetNet topology with low traffic load) were averaged. The same sequences of experiment condition are run for each routing algorithm. In the upcoming section, the comparison of the RL-based framework against the other routing algorithms (MHA, WSP, SWP and MIRA) is presented with respect to the impact on the topology level and traffic load. Thus, the four routing algorithms are applied individually and remained fixed for routing both QoS-based and background flows during the experiment. The RL-based framework dynamically selects a routing algorithm for the HD live video traffic each time when a new state is monitored, while the MIRA algorithm is kept static to route all flows belonging to SD video, HTTP and FTP traffic.

### A. Impact of Traffic load

This section presents the impact of the traffic load on the performance of the proposed RL-based framework as compared to other routing algorithms under different topologies. It shows the performance comparison across various network loads taking into account the same topology.

*1) GetNet Topology:* The results for the small scale network like GetNet topology are illustrated in Tables V, VI, and VII and Figures 3 and 4. The results in Table V and Figure 3 show that all routing algorithm including the RL-based framework produce the highest throughput and lower packet loss for all types of traffic services under the low traffic load. For example, all solutions get throughput of 650Kb/s and packet loss between 0.05-0.1% for the QoS-based traffic. Similar trend of results is observed when the level of traffic load increases to medium. In general, the QoS-based traffic flows meet the requirements given by Table IV. This can be attributed to the fact that the algorithms route the newly coming flows efficiently while rejecting those that are causing the network congestion.

As the load increases to high, the network gets congested and the traffic flows experience packet losses. However, the results show that the proposed RL-based solution performs

TABLE VI
AVERAGED LATENCY EVALUATION FOR ROUTING ALGORITHMS UNDER GETNET, SPRINT AND AT&T, WHERE L = LOW LOAD, M = MEDIUM LOAD, AND H = HIGH LOAD

| | MHA Latency [ms] | | | WSP Latency [ms] | | | SWP Latency [ms] | | | MIRA Latency [ms] | | | RL-based Method Latency [ms] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | l | m | h | l | m | h | l | m | h | l | m | h | l | m | h |
| **GetNet** | | | | | | | | | | | | | | | |
| HD | 7.5 | 20 | 1379 | 9 | 90 | 1320 | 13 | 219 | 1758 | 9 | 91 | 1214 | 8 | 18 | 900 |
| SD | 3 | 34 | 314 | 3 | 143 | 127 | 4 | 189 | 424 | 4 | 142 | 76 | 3 | 59 | 46 |
| HTTP | 2 | 9 | 81 | 2 | 19 | 123 | 3 | 55 | 235 | 2 | 19 | 84 | 2 | 18 | 28 |
| FTP | 2 | 30 | 245 | 2 | 67 | 17 | 3 | 230 | 354 | 2 | 68 | 33 | 2 | 41 | 25 |
| **Sprint** | | | | | | | | | | | | | | | |
| HD | 84 | 662 | 838 | 70 | 754 | 835 | 27 | 633 | 782 | 66 | 650 | 739 | 60 | 47 | 78 |
| SD | 38 | 60 | 66 | 27 | 73 | 124 | 14 | 61 | 66 | 38 | 26 | 75 | 32 | 51 | 50 |
| HTTP | 17 | 52 | 92 | 19 | 43 | 119 | 9 | 73 | 80 | 12 | 29 | 67 | 15 | 25 | 30 |
| FTP | 72 | 62 | 74 | 62 | 65 | 118 | 20 | 124 | 130 | 46 | 39 | 83 | 51 | 46 | 52 |
| **AT&T** | | | | | | | | | | | | | | | |
| HD | 523 | 675 | 687 | 483 | 602 | 589 | 604 | 399 | 292 | 571 | 708 | 613 | 141 | 191 | 265 |
| SD | 28 | 40 | 54 | 16 | 90 | 43 | 100 | 105 | 45 | 23 | 81 | 39 | 17 | 48 | 44 |
| HTTP | 40 | 58 | 70 | 14 | 50 | 41 | 83 | 70 | 30.5 | 22 | 42 | 50 | 13 | 22 | 21 |
| FTP | 67 | 108 | 112 | 18 | 101 | 72 | 124 | 141 | 52 | 22 | 88 | 81 | 33 | 44 | 38 |

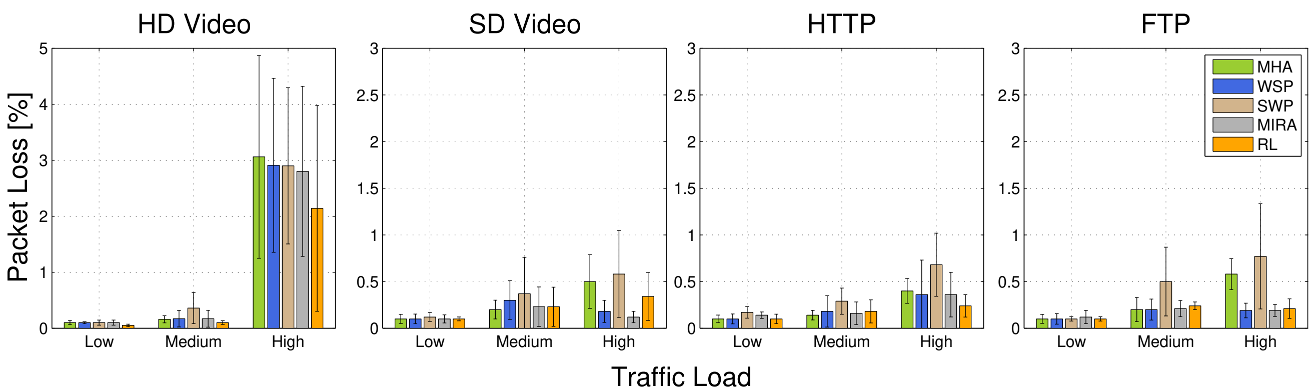



Fig. 3. GetNet network topology: Packet loss of the traffic classes under different traffic loads

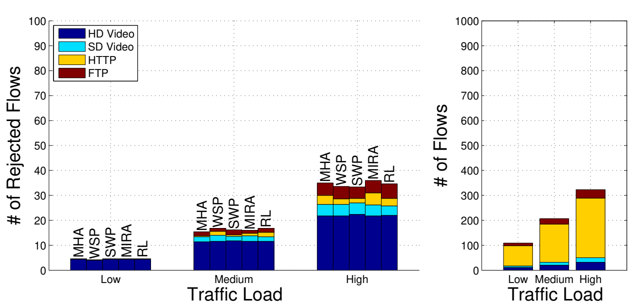



Fig. 4. GetNet network topology: The total number of rejected flow and the total number of flows that are generated in the experiment test

better in terms of throughput, packet loss and PSNR when compared to other routing algorithms on their own. With the RL-based solution, the packet loss for QoS-based service reaches 2.14% as compared to MHA algorithm with 3.06% (as shown in Figure 3). Consequently, it can be seen that under the small scale network, all the solutions maintain an *Excellent* QoE (see Table VII) for the QoS-based services when the traffic load increases from low to medium. However, under high traffic load only the RL-based method maintains a *Good* QoE, while the other solutions drop the user perceived quality for QoS-based services to *Fair*.

On the other hand, Figure 4 shows the number of rejected flows for each solution. As expected, the results show that the flow rejection of QoS-based traffic grows with the increase in network load. This is due to the throughput of HD video and the high arrival rate of new flows. This in general leads to more QoS-based flows to be rejected. In other words, the routing algorithm cannot allocate a path to the newly coming flows on the network. For example at high traffic load, MHA, WSP, SWP, MIRA,

TABLE VII
AVERAGED ESTIMATED PSNR AND MOS EVALUATION FOR THE ROUTING ALGORITHMS UNDER GETNET, SPRINT AND AT&T, WHERE L = LOW LOAD, M = MEDIUM LOAD, AND H = HIGH LOAD

| | | MHA | | | WSP | | | SWP | | | MIRA | | | RL-based Method | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | l | m | h | l | m | h | l | m | h | l | m | h | l | m | h |
| **GetNet** | | | | | | | | | | | | | | | | |
| HD | PSNR [dB] | 60 | 55.9 | 30.3 | 60 | 55.4 | 30.7 | 60 | 48.9 | 30.7 | 60 | 55.4 | 31.1 | 66 | 60 | 33.4 |
| | MOS | Exc. | Exc. | Fair | Exc. | Exc. | Fair | Exc. | Exc. | Fair | Exc. | Exc. | Fair | Exc. | Exc. | Good |
| SD | PSNR [dB] | 60 | 53.9 | 46 | 60 | 50.5 | 54.9 | 58.4 | 48.6 | 44.7 | 60 | 52.8 | 58.4 | 60 | 52.8 | 49.4 |
| | MOS | Exc. | Exc. | Exc. | Exc. | Exc. | Exc. | Exc. | Exc. | Good | Exc. | Exc. | Exc. | Exc. | Exc. | Exc. |
| **Sprint** | | | | | | | | | | | | | | | | |
| 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 | 52 | 49.6 | 46 |
| | MOS | Exc. | Poor | Poor | Exc. | Poor | Poor | Exc. | Poor | Poor | Exc. | Poor | Poor | Exc. | Exc. | Exc. |
| 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 | 52.3 | 49.1 | 49.8 |
| | MOS | Exc. | Exc. | Exc. | Exc. | Exc. | Good | Exc. | Good | Exc. | Exc. | Exc. | Exc. | Exc. | Exc. | Exc. |
| **AT&T** | | | | | | | | | | | | | | | | |
| 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 |

and the proposed RL-based approach achieve an HD video flow rejection of 21.8, 21.8, 22.4, 21.8, and 22, respectively, from a total traffic volume of 32 flows. Compared to the RL-based solution, all other routing algorithms attempt to accommodate more QoS-based flows even under high load, at the cost of decreasing the users' QoE to *Fair*, while the RL-based method finds the best trade-off between the throughput, packet loss and rejection rate and maintains a *Good* QoE for the QoS-based services without sacrificing the other traffic classes either.

*2) Sprint Topology:* The results for the medium scale network like Sprint topology are illustrated in Tables V, VI, and VII and Figures 5 and 6. Table V and Figure 5 show that all algorithms under the low traffic load perform similarly with low packet loss and high throughput measurements. This is due to the fact that the algorithms successfully find a path for the incoming flows under a low number of generated traffic flows and reject the flows that are causing link congestion. For example, MHA, WSP, SWP, MIRA, and the proposed RL-based solution get low average packet loss of 0.3%, 0.27%, 0.13%, 0.25%, and 0.25% respectively. However, as the network load increases to medium, the proposed RL-based method starts to outperform other routing algorithms. The RL-based method shows better results with 0.33% packet loss for QoS-based traffic as compared to other routing algorithms such as MIRA with a packet loss of 3.83%. Similarly, the RL-based method outperforms MIRA algorithm in terms of PSNR by an increase in the estimated averaged PSNR of 21.3dB (as shown in Table VII). Thus, there is a considerable decrease in packet loss when the RL-based method is applied. For example, the RL-based method shows a packet loss of 0.33%, while MHA, WSP, SWP, MIRA get an average packet loss of 4.68%, 4.57%, 4.06%, and 3.83%, respectively. In terms of maximizing throughput for QoS-based services, it is observed that on average the RL-based algorithm outperforms others by achieving 643Kb/s throughput and latency of 47ms. In particular, in the Sprint topology the classical routing algorithms suffer from an significant increase in packet loss when the network load increases from low to high, while the RL-based solution shows better results in this respect. For example, the RL-based method shows an increase of only 0.25% from low to high load.

Thus, as the network load increases to high, the packet loss for the classical routing algorithms increases significantly while the RL-based method makes a significant improvement in terms of minimizing the packet loss by reaching 0.5% only. For example, MHA algorithm achieves packet loss of 6.76%. When looking at maximizing the throughput and minimizing the packet loss for HD video, the RL-based method achieves better results compared to other algorithms under medium and high loads. For example, even when the traffic load increases from low to high, the RL-based solution finds the best trade-off between the throughput, packet loss and rejection rate and maintains an *Excellent* user perceived QoE (as per Table III and Table VII) for the QoS-based traffic class, without penalizing the other traffic classes. However, in the case of all other routing algorithms as the traffic load increases, the routing algorithms try to accommodate more QoS-based flows at the cost of a severe degradation in the user perceived QoE, by dropping the MOS from *Excellent* under low load to *Poor* under high load. Consequently, the QoS requirements for the QoS-based traffic class are not met.

*3) AT&T Topology:* The results for the large scale network like AT&T topology are illustrated in Tables V, VI, and VII

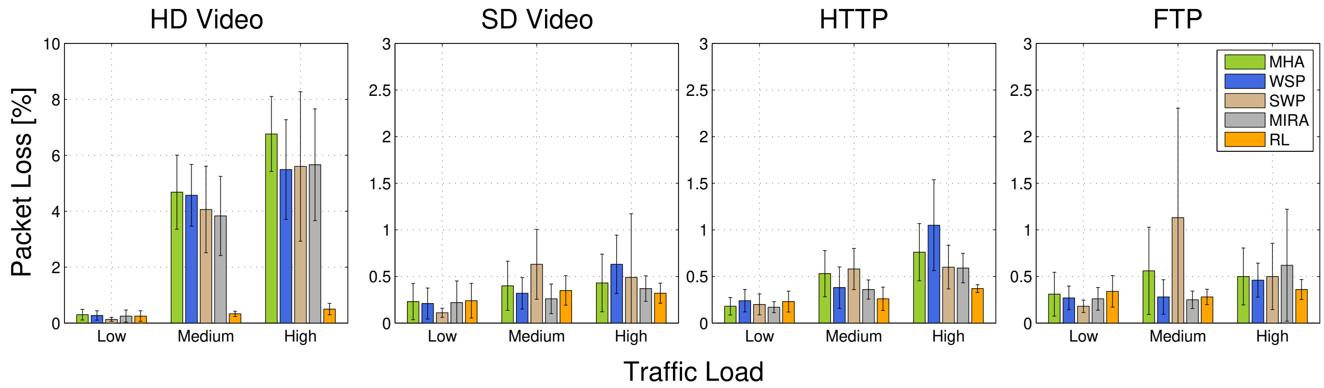



Fig. 5. Sprint network topology: Packet loss of the traffic classes under different traffic loads

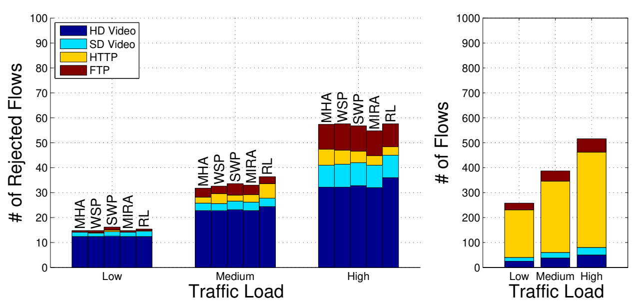



Fig. 6. Sprint network topology: The total number of rejected flow and the total number of flows that are generated in the experiment test

and Figures 7 and 8. While looking at the results within the AT&T network topology but under different traffic loads, it can be observed that on average, the proposed RL-based method outperforms other routing algorithms with respect to the HD video traffic. Table V and Figure 7 show that all routing algorithms reach lower throughput and higher packet loss when compared to the proposed RL-based method. For example, under low traffic load, the RL-based algorithm reaches 1.07% packet loss for the QoS-based services while MHA, WSP, SWP, MIRA achieved an average packet loss of 5.34%, 4.09%, 7.03%, and 4.47% respectively. As seen in Table VII, this translates in an estimated averaged PSNR of 39.4dB for the proposed RL-based method. Thus, the RL-based method makes a significant improvement in terms of minimizing the packet loss when compared to the classical routing algorithms.

Figure 8 shows that all solutions lead to rejecting more of the incoming flows of QoS-based traffic class when the network load increases. Due to the increase in the total amount of generated video traffic while the network capacity stays fixed, the flow rejection rate of the QoS-based services becomes implicitly higher.

In particular, it is noticeable that the proposed RL-based method draws advantages when applied on a large scale network. It outperforms other classical routing algorithms in terms of maximizing throughput and minimizing the packet loss when the network load increases from low to high. Even under a large-scale network, the proposed RL-based solution provides a *Good* (see Table III) 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.

### *B. Impact of Network Topology*

This section studies the impact of the network topology on the performance of the RL-based framework based on the traffic load level. It shows the performance comparison of various solutions across the network topologies taking into account the load level.

*1) Low Traffic Load:* As depicted in Figures 3, 5, and 7, it can be seen that the proposed RL-based method shows relatively a lower packet loss in the range of 0.05% and 1.07% under various network topologies (i.e. GetNet, Sprint, and AT&T). By contrast, looking at other classical routing algorithms, it can be observed that they experience higher packet loss as the topology increases from small to large scale network. For example, as the network size increases from GetNet to AT&T, the packet loss of HD video under the proposed RL-based method only increased by 1.02% on average as compared to MHA that has an increase of 5.24%. This comparison also corresponds to a smaller decrease in the estimated averaged PSNR of the QoS-based video service by 26.6dB for the proposed RL-based method. Whereas MHA showed a larger drop by 34.6dB (see Table VII). Similarly, in terms of minimizing the latency for QoS-based flows, the proposed RL-based method performs better than other classical routing algorithms. For instance, MIRA produces on an average four times higher delay than the RL-based method for AT&T large scale network (see Table VI). The results show that under low traffic load, as the network size increases, the proposed RL-based solution

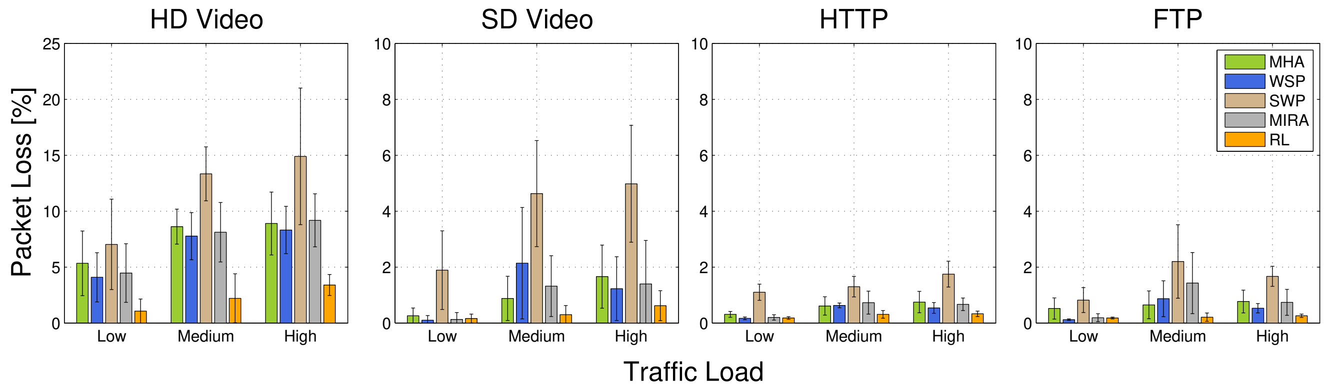



Fig. 7. AT&T network topology: Packet loss of the traffic classes under different traffic loads

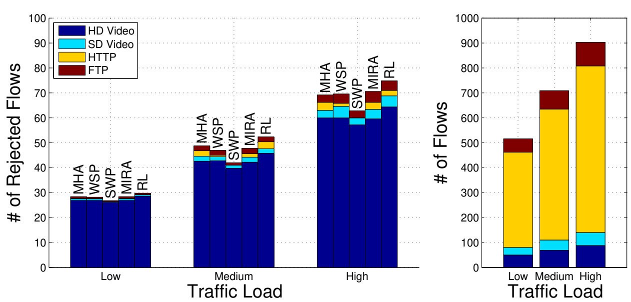



Fig. 8. AT&T network topology: The total number of rejected flow and the total number of flows that are generated in the experiment test

decides the most suitable routing algorithms to be applied at each decision interval, in such a way that, the QoS requirements for the QoS-based traffic class are met while the other traffic classes are not penalized.

Figures 4, 6, and 8 show on average the number of rejected flows. The rejected flows represent the newly incoming flows that cannot be allocated a path on the network, because the network reaches a level where the links are mostly utilized. In terms of the flow rejection of QoS-based services, the figures show that the classical routing algorithms produce lower flow rejection of the QoS-based service when compared to the RL-based method. Consequently, even if the network size increases, e.g., AT&T under low load, the RL-based method maintains a *Good* QoE for the QoS-based traffic as seen in Table VII, while the other solutions sacrifice the QoE of the QoS-based traffic by dropping the MOS to *Poor* (e.g., MHA, SWP) and *Fair* (e.g., WSP, MIRA) in an attempt to accommodate more QoS-based flows.

From the results in Table V and the Figures 3, 6 and 8, it is noticeable that the background traffic (i.e., SD video, FTP and HTTP) achieve a considerably good overall performance. Referring to the requirements set in Table IV, the earlier results demonstrate that the proposed RL-based dynamic routing algorithm meets on average the requirement of the QoS-based traffic under low traffic load over the GetNet, Sprint and AT&T topologies.

2) *Medium Traffic Load:* For the medium load, the results in Figures 3, 5, and 7 illustrate that on average the proposed RL-based algorithm outperforms other routing algorithms in terms of maximizing the throughput and minimizing the packet loss and latency while maintaining an acceptable user perceived quality for the QoS-based services without penalizing the other traffic classes. As the network size increases from GetNet to AT&T, the results show that other routing algorithms (i.e., MHA, WSP, SWP and MIRA) experience higher packet loss by an increase of 8.46%, 7.6%, 12.98%, and 7.95%, respectively. While the proposed RL-based method achieves considerably better results with an increase of packet loss by 2.11% for the quality service. Likewise, when looking at maximizing the throughput for the QoS-based video flows, the RL-based method performs better by achieving 630Kb/s throughput under AT&T large-scale network as compared to WSP with a throughput of 604Kb/s (see Table V). Similar trend is observed by the RL-based method in terms of latency and PSNR. For example, as the network size increases from GetNet to AT&T, the RL-based method implies a small decrease in PSNR of QoS-based video service by 26.9dB, while there is a larger decrease for MHA, WSP, SWP and MIRA algorithms by 34.6, 33.2, 31.4, and 33.6dB, respectively. It is important to be noted that the policy of the RL-based method is trained to maximize the expected cumulative long-term reward in terms of throughput and packet loss. Based on the trained policy, at each time-slot of the monitoring cycle, the algorithm chooses the best candidate of the routing algorithm that suits the actual state. This can significantly reduce loss rates across congested links by rerouting the actual traffic flows accordingly.

With respect to the number of rejections for the QoS-based flows, it can be observed in Figures 4, 6, and 8 that in general, the number of flow rejections for the HD video grows with the increase in the network size under the same network load. However, the classical routing algorithms exhibits lower number of rejections than the RL-based method. Accommodating more flows comes at the cost of decrease in user perceived quality. For example, under medium load, the proposed RL-based method maintains an *Excellent* (e.g., GetNet, Sprint) to *Good* (e.g., AT&T) QoE (see Table VII) as the network size increases, while the other solutions drop the QoE from *Excellent* (e.g., all routing algorithms over GetNet) to *Fair* (e.g., SWP and MIRA over Sprint), *Poor* (e.g., MHA and WSP over Sprint and AT&T, MIRA over AT&T) and even *Bad* (e.g., SWP over AT&T). Consequently, the proposed RL-based method finds the best trade-off between the throughput, packet loss and the rejection rate so that it maintains an acceptable user perceived quality for the QoS-based traffic even if the network size increases, without sacrificing the other traffic classes. All classical routing algorithms are showing difficulty in accommodating good performance across the topology changes. This is due to the fact that the classical routing algorithm cannot react to the network changes when newly coming flows arrive within the time-slot of the monitoring interval. On the contrary, the RL-based method initiates a rerouting mechanism at every time-slot of monitoring interval and it applies the optimal routing on the actual network state. According to the QoS requirements stated in Table IV, the RL-based method conforms to the requirement under the GetNet and Sprint network. Though, under the AT&T network, the results in Figure 7 show that the RL-based method deviates from the given requirement by only 1.21% in the average packet loss. The other classical routing algorithms show higher deviations from the requirements under the medium and large scale networks. For example, above 3% under the Sprint network, while it becomes higher than 7% in the case of AT&T network.

*3) High Traffic Load:* Figures 3, 5 and 7 reveal that there is a variation in performance for the RL-based method under the high load with the increase in the network topology size. It can be noted an increase in packet loss of 1.26% and 2.9% when the network size increases from GetNet to AT&T and from Sprint to AT&T, respectively. The increase from GetNet to Sprint topology leads to a decrease in the packet loss of 1.64%. On one hand, these results are due to the fact that the GetNet network is small and the proposed RL-based method can not resolve the network congestion by rerouting the traffic flows. On the other hand, it can be observed that the classical routing algorithms exhibit higher packet loss in larger network than in smaller network. For example, as the network topology increases from GetNet to AT&T, the packet loss for MIRA algorithm is increased considerably by 6.38% when compared to the RL-based method of only 1.26%.

In terms of rejection rate, the RL-based method has a higher rejection of flows in order to avoid packet loss caused by network overload and drop in QoE. Consequently, it can be noticed that under high load, as the network size increases, the RL-based method still maintains an *Excellent* (e.g., Sprint), *Good* (e.g., GetNet) and *Fair* (e.g., AT&T) user perceived quality (see Table VII) for the QoS-based service without penalizing the other traffic classes. In contrast, the other solutions are only able to maintain a *Fair* QoE under GetNet, and as the network topology size increases, this drops to *Poor* for all the other scenarios except for SWP over AT&T where it drops to *Bad*. Thus, the size of the network determines the performance of the routing algorithms under the high load. Though in general, the results of the proposed RL-based method are very promising when compared to the classical routing algorithms.

### C. Impact of the Background Traffic Routing Algorithm

In order to validate the choice of the routing algorithm for the background traffic (e.g., MIRA), a set of experiments have been conducted that compare the system performance when using any of the four routing algorithms as a choice for routing the background traffic. For the validation purposes, only the Sprint topology under the three network loads was used. The scenarios were evaluated under the same environment conditions (e.g. the total number of generated flows) and they were averaged over 5 simulation trails for each single scenario. The results are illustrated in Table VIII. It can be seen that all routing algorithms exhibit relatively similar averaged performance under various traffic loads. Therefore, because the routing algorithm MIRA performs generally well at the network level, it was selected to route the background flows when employing the RL-based approach.

## VII. Conclusions

This paper introduces an innovative Reinforcement Learning (RL)-based framework for multimedia-based SDN environments. The proposed RL-based framework makes use of ML to decide on the most suitable routing algorithm to be applied on the QoS-based traffic flows within a realistic multimedia-based SDN environment in order to enable QoS provisioning. The proposed RL-based solution was implemented and evaluated using an experimental setup based on Mininet, Floodlight controller and Open vSwitch switches. The results show that the proposed RL-based solution outperforms the other state-of-the-art routing algorithms (i.e., MHA, WSP, SWP and MIRA) and finds the best trade-off between throughput, packet loss and rejection rate for the QoS-based traffic class without penalizing the other traffic classes. Even under the largest network topology (e.g., AT&T) and highest traffic load, the RL-based solution ensures a *Fair* user perceived QoE for the QoS-based services while all the other solutions will significantly degrade the user perceived QoE to *Poor* in the case of MHA, WSP, and MIRA and *Bad* in the case of SWP in an attempt to accommodate more QoS-based flows as well as the other traffic classes are penalized with increased packet loss rate. Under all the other considered scenarios, the proposed RL-based method maintains an *Excellent* to *Good* user perceived QoE, while all the other state-of-the-art go as low as *Poor* in case of MHA, WSP and MIRA, and even *Bad*, in case of SWP.

## References

[1] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, and et al., "The lockdown effect," *Proceedings of the ACM Internet Measurement Conference*, Oct 2020.

[2] 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*, vol. 20, no. 2, pp. 945–977, 2018.

TABLE VIII
SPRINT NETWORK TOPOLOGY: AVERAGED PERFORMANCE EVALUATION FOR THE RL ROUTING ALGORITHMS (REROUTING THE QOS-BASED TRAFFIC USING THE RL-BASED METHOD, WHILE THE ROUTING OF BACKGROUND TRAFFIC IS BASED ON THE STATIC ALGORITHM)

| | **Performance Metrics** | QoS flows ⇒ RL, Bkg flows ⇒ MHA | | | QoS flows ⇒ RL, Bkg flows ⇒ WSP | | | QoS flows ⇒ RL, Bkg flows ⇒ SWP | | | QoS flows ⇒ RL, Bkg flows ⇒ MIRA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | l | m | h | l | m | h | l | m | h | l | m | h |
| HD | Throughput [Kb/s] | 646 | 644 | 630 | 647 | 645 | 639 | 644 | 645 | 635 | 646 | 643 | 638 |
| | Packet Loss [%] | 0.23 | 0.52 | 0.75 | 0.2 | 0.42 | 0.52 | 0.23 | 0.4 | 0.46 | 0.25 | 0.33 | 0.5 |
| | Latency [ms] | 58 | 98 | 105 | 42 | 72 | 75 | 46 | 52 | 77 | 60 | 47 | 78 |
| | Estimated PSNR [dB] | 52.8 | 45.7 | 42.5 | 54 | 47.5 | 45.7 | 52.8 | 47.9 | 46.7 | 52 | 49.6 | 46 |
| | # of Rejected Flows | 12.4 | 25 | 36 | 12.2 | 25.4 | 35 | 12.2 | 24.6 | 36.8 | 12.4 | 24.4 | 36 |
| SD | Throughput [Kb/s] | 236 | 235 | 235 | 236 | 235 | 234 | 236 | 235 | 234 | 236 | 234 | 235 |
| | Packet Loss [%] | 0.24 | 0.27 | 0.26 | 0.18 | 0.33 | 0.36 | 0.19 | 0.23 | 0.46 | 0.24 | 0.35 | 0.32 |
| | Latency [ms] | 33 | 36 | 35 | 22 | 49 | 27 | 19 | 31 | 41 | 32 | 51 | 50 |
| | Estimated PSNR [dB] | 52.4 | 51.4 | 51.7 | 54.9 | 49.6 | 48.9 | 54.4 | 52.7 | 46.7 | 52.3 | 49.1 | 49.8 |
| | # of Rejected Flows | 1.2 | 3.6 | 7.2 | 1 | 4.2 | 8.4 | 1.4 | 4.8 | 7.8 | 2.2 | 3.4 | 9 |
| HTTP | Throughput [Kb/s] | 15.6 | 16.0 | 16.7 | 15.4 | 16.3 | 16.7 | 15.5 | 17.1 | 16.5 | 15.3 | 15.9 | 14.3 |
| | Packet Loss [%] | 0.23 | 0.31 | 0.34 | 0.25 | 0.25 | 0.45 | 0.2 | 0.38 | 0.48 | 0.23 | 0.26 | 0.37 |
| | Latency [ms] | 14 | 24 | 20 | 14 | 26 | 23 | 11 | 24 | 29 | 15 | 25 | 30 |
| | # of Rejected Flows | 0.4 | 2.4 | 5.2 | 0.4 | 4 | 3.6 | 0 | 2.2 | 6.8 | 0.2 | 5.8 | 3.4 |
| FTP | Throughput [Kb/s] | 187 | 177 | 169 | 188 | 177 | 169 | 188 | 178 | 171 | 186.3 | 177.8 | 167.9 |
| | Packet Loss [%] | 0.25 | 0.45 | 0.48 | 0.22 | 0.32 | 0.38 | 0.19 | 0.39 | 0.4 | 0.34 | 0.28 | 0.36 |
| | Latency [ms] | 35 | 43 | 65 | 24 | 55.6 | 58 | 14 | 41 | 48 | 51 | 46 | 52 |
| | # of Rejected Flows | 1 | 1.4 | 8.8 | 0.8 | 3.2 | 10.2 | 0.6 | 3.2 | 9 | 0.6 | 2.8 | 9.2 |

[3] A. Doumanoglou, N. Zioulis, D. Griffin, J. Serrano, T. K. Phan, D. Jiménez, D. Zarpalas, F. Álvarez, M. Rio, and P. Daras, “A system architecture for live immersive 3D-media transcoding over 5G networks,” in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2018, pp. 11–15.

[4] N. Jawad, M. Salih, K. Ali, B. Meunier, Y. Zhang, X. Zhang, R. Zetik, C. Zarakovitis, H. Koumaras, M. Kourtis, L. Shi, W. Mazurczyk, and J. Cosmas, “Smart television services using NFV/SDN network management,” *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 404–413, June 2019.

[5] L. Fay, L. Michael, D. Gómez-Barquero, N. Ammar, and M. W. Caldwell, “An overview of the ATSC 3.0 physical layer specification,” *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 159–171, March 2016.

[6] C. Ge, N. Wang, I. Selinis, J. Cahill, M. Kavanagh, K. Liolis, C. Politis, J. Nunes, B. Evans, Y. Rahulan, N. Nouvel, M. Boutin, J. Desmauts, F. Arnal, S. Watts, and G. Poziopoulou, “QoE-assured live streaming via satellite backhaul in 5G networks,” *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 381–391, June 2019.

[7] R. Viola, A. Martin, M. Zorrilla, and J. Montalbán, “MEC proxy for efficient cache and reliable multi-CDN video distribution,” in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2018, pp. 1–7.

[8] B. Li, Z. Fei, and Y. Zhang, “UAV communications for 5G and beyond: Recent advances and future trends,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, April 2019.

[9] A. Martin, J. Egaña, J. Flórez, J. Montalbán, I. G. Olaizola, M. Quartulli, R. Viola, and M. Zorrilla, “Network resource allocation system for QoE-aware delivery of media services in 5G networks,” *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 561–574, June 2018.

[10] I.-S. Comsa, G. Muntean, and R. Trestian, “An innovative machine-learning-based scheduling solution for improving live uhd video streaming quality in highly dynamic network environments,” *IEEE Transactions on Broadcasting*, pp. 1–13, 2020.

[11] W. Braun and M. Menth, “Software-defined networking using openflow: Protocols, applications and architectural design choices,” *Future Internet*, vol. 6, no. 2, pp. 302–336, 2014.

[12] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, “Applications of artificial intelligence and machine learning in smart cities,” *Computer Communications*, vol. 154, pp. 313 – 323, 2020.

[13] Z. Ullah, F. Al-Turjman, U. Moatasim, L. Mostarda, and R. Gagliardi, “Uavs joint optimization problems and machine learning to improve the 5g and beyond communication,” *Computer Networks*, vol. 182, p. 107478, 2020.

[14] S. Mihai, P. Shah, G. Mapp, H. Nguyen, and R. Trestian, “Towards autonomous driving: A machine learning-based pedestrian detection system using 16-layer lidar,” in *2020 13th International Conference on Communications (COMM)*, 2020, pp. 271–276.

[15] S. Norden, “Inter-domain routing: Algorithms for qos guarantees,” *Computer Networks*, vol. 49, no. 4, pp. 593–619, 2005.

[16] C. Pornavalai, G. Chakraborty, and N. Shiratori, “Qos based routing algorithm in integrated services packet networks,” *Journal of High Speed Networks*, vol. 7, no. 2, pp. 99–112, 1998.

[17] G. Araniti, J. Cosmas, A. Iera, A. Molinaro, R. Morabito, and A. Orsino, “Openflow over wireless networks: Performance analysis,” in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. IEEE, 2014, pp. 1–5.

[18] S. Tomovic, I. Radusinovic, and N. Prasad, “Performance comparison of qos routing algorithms applicable to large-scale sdn networks,” in *EUROCON 2015-International Conference on Computer as a Tool (EUROCON), IEEE*. IEEE, 2015, pp. 1–6.

[19] M.-C. Lee and J.-P. Sheu, “An efficient routing algorithm based on segment routing in software-defined networking,” *Computer Networks*, vol. 103, pp. 44–55, 2016.

[20] S. Abdallah, A. Kayssi, I. H. Elhajj, and A. Chehab, “Network convergence in sdn versus ospf networks,” in *2018 Fifth International Conference on Software Defined Systems (SDS)*. IEEE, 2018, pp. 130–137.

[21] H. E. Egilmez, S. T. Dane, B. Gorkemli, and A. M. Tekalp, “Openqos: Openflow controller design and test network for multimedia delivery with quality of service,” *Proc. NEM Summit, Implementing Future Media Internet Towards New Horizons*, pp. 22–27, 2012.

[22] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, “An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks,” *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 710–715, 2013.

[23] H. E. Egilmez, B. Gorkemli, A. M. Tekalp, and S. Civanlar, "Scalable video streaming over openflow networks: An optimization framework for qos routing," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2241–2244.

[24] T.-F. Yu, K. Wang, and Y.-H. Hsu, "Adaptive routing for video streaming with QoS support over SDN networks," in *Information Networking (ICOIN), 2015 International Conference on*. IEEE, 2015, pp. 318–323.

[25] F. Ongaro, E. Cerqueira, L. Foschini, A. Corradi, and M. Gerla, "Enhancing the quality level support for real-time multimedia applications in software-defined networks," in *Computing, Networking and Communications (ICNC), 2015 International Conference on*. IEEE, 2015, pp. 505–509.

[26] X. Jin, H. Ju, S. Cho, B. Mun, C. Kim, and S. Han, "Qos routing design for adaptive streaming in software defined network," in *Intelligent Signal Processing and Communication Systems (ISPACS), 2016 International Symposium on*. IEEE, 2016, pp. 1–6.

[27] Q. Duan, C. Wang, and X. Li, "End-to-end service delivery with qos guarantee in software defined networks," *arXiv preprint arXiv:1504.04076*, 2015.

[28] S. E. Middleton and S. Modafferi, "Experiences monitoring and managing qos using sdn on testbeds supporting different innovation stages," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–5.

[29] Y. Oh, J. Kim, and J. Lee, "Qos constrained path optimization algorithm in nfv/sdn environment," *ICN 2016*, p. 142, 2016.

[30] S. Layeghy, F. Pakzad, and M. Portmann, "Scor: Software-defined constrained optimal routing platform for sdn," *arXiv preprint arXiv:1607.03243*, 2016.

[31] S. Tomovic and I. Radusinovic, "A new traffic engineering approach for qos provisioning and failure recovery in sdn-based isp networks," in *2018 23rd International Scientific-Professional Conference on Information Technology (IT)*. IEEE, 2018, pp. 1–4.

[32] H. Ghalwash and C.-H. Huang, "A qos framework for sdn-based networks," in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2018, pp. 98–105.

[33] S. J. Y. Go, C. A. M. Festin, and W. M. Tan, "An sdn-based framework for improving the performance of underprovisioned ip video surveillance networks," *Journal of Network and Computer Applications*, vol. 132, pp. 49–74, 2019.

[34] S. Zhu, Z. Sun, Y. Lu, L. Zhang, Y. Wei, and G. Min, "Centralized qos routing using network calculus for sdn-based streaming media networks," *IEEE Access*, vol. 7, pp. 146 566–146 576, 2019.

[35] S. Ren, W. Dou, and Y. Wang, "A deterministic network calculus enabled qos routing on software defined network," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2017, pp. 182–186.

[36] P. Tantisarkhornkhet and W. Werapun, "Qlb: Qos routing algorithm for software-defined networking," in *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2016, pp. 1–6.

[37] D. Adami, G. Antichi, R. G. Garroppo, S. Giordano, and A. W. Moore, "Towards an sdn network control application for differentiated traffic routing," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5827–5832.

[38] J. Liu and Q. Xu, "Machine learning in software defined network," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, 2019, pp. 1114–1120.

[39] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2018.

[40] A. A. Gebremariam, M. Usman, and M. Qaraqe, "Applications of artificial intelligence and machine learning in the area of sdn and nfv: A survey," in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2019, pp. 545–549.

[41] M. Latah and L. Toker, "Artificial intelligence enabled software-defined networking: a comprehensive overview," *IET Networks*, vol. 8, no. 2, pp. 79–99, 2018.

[42] ——, "Application of artificial intelligence to software defined networking: A survey," *Indian Journal of Science and Technology*, vol. 9, no. 44, 2016.

[43] J. Qadir, K.-L. A. Yau, M. A. Imran, Q. Ni, and A. V. Vasilakos, "Ieee access special section editorial: Artificial intelligence enabled networking," *IEEE Access*, vol. 3, pp. 3079–3082, 2015.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[50] 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.

[51] 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.

[52] G. Stampa, M. Arias, D. Sánchez-Charles, V. Muntés-Mulero, and A. Cabellos, "A deep-reinforcement learning approach for software-defined networking routing optimization," *arXiv preprint arXiv:1709.07080*, 2017.

[53] A. Al-Jawad, P. Shah, O. Gemikonakli, I. S. Comşa, and R. Trestian, "Performance evaluation of routing strategies over multimedia-based sdns under realistic environments," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, 2020, pp. 160–164.

[54] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[55] M. Parashar and S. Hariri, "Autonomic computing: An overview," in *International workshop on unconventional programming paradigms*. Springer, 2004, pp. 257–269.

[56] 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.

[57] 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.

[58] J. F. Kurose, *Computer networking: A top-down approach featuring the internet, 3/E*. Pearson Education India, 2005.

[59] A. Hassidim, D. Raz, M. Segalov, and A. Shaqed, "Network utilization: The flow view," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1429–1437.

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

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

[62] M. Ponsen, M. E. Taylor, and K. Tuyls, "Abstraction and generalization in reinforcement learning: A summary and framework," in *International Workshop on Adaptive and Learning Agents*. Springer, 2009, pp. 1–32.

[63] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

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

[65] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[66] 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.

[67] Mininet openflow virtual network. Accessed on August 1, 2015. [Online]. Available: http://mininet.org

[68] Floodlight openflow controller. Accessed on August 1, 2015. [Online]. Available: http://www.projectfloodlight.org/

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

[70] 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.

[71] "FFMPEG video and audio converter tool," https://ffmpeg.org/, accessed on Aug 1, 2020.

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

[73] 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.

[74] H.-K. Choi and J. O. Limb, "A behavioral model of web traffic," in *Network Protocols, 1999.(ICNP'99) Proceedings. Seventh International Conference on*. IEEE, 1999, pp. 327–334.

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

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

[77] A. S. Tannenbaum, *Computer networks*. Pearson Education India, 2002.

[78] Y. L. Lee, J. Loo, and T. C. Chuah, "Modeling and performance evaluation of resource allocation for lte femtocell networks," in *Modeling and Simulation of Computer Networks and Systems*. Elsevier, 2015, pp. 683–716.

[79] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[80] 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.

[81] P. ITU-T RECOMMENDATION, "Subjective video quality assessment methods for multimedia applications," *International telecommunication union*, 1999.

[82] S. A. Hussain, *Active and programmable networks for adaptive architectures and services*. CRC Press, 2006.

[83] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*. Elsevier, 2004.

[84] H. Zhai, X. Chen, and Y. Fang, "How well can the ieee 802.11 wireless lan support quality of service?" *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 3084–3094, 2005.

[85] I. Union, "Itu-t g. 1010: End-user multimedia qos categories," *G SERIES: Transmission Systems and Media, Digital System and Networks-Multimedia Quality of Service and Performance Generic and User-Related Aspects,*, 2001.

[86] U. Lakshman and L. Lobo, *MPLS configuration on Cisco IOS software*. Cisco Press, 2005.

[87] S. P. Chris Lewis, *Implementing Quality of Service Over Cisco MPLS VPNs*. Cisco Press, 2006.

[88] R. A. Cacheda, D. C. García, A. Cuevas, F. J. G. Castano, J. H. Sánchez, G. Koltsidas, V. Mancuso, J. I. M. Novella, S. Oh, and A. Pantò, "Qos requirements for multimedia services," in *Resource management in satellite networks*. Springer, 2007, pp. 67–94.

[89] 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/

[90] T. Szigeti, C. Hattingh, R. Barton, and K. Briley Jr, *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. Cisco Press, 2013.

[91] W. Matthews, "Internet monitoring in the hep community," Stanford Linear Accelerator Center, Menlo Park, CA (US), Tech. Rep., 1998.

[92] F. Farid, "Methodical evaluation of quality of service for heterogeneous networks," *PhD Thesis,*, 2015.

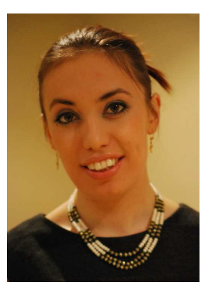
[93] S.-B. Lee, G.-M. Muntean, and A. F. Smeaton, "Performance-aware replication of distributed pre-recorded iptv content," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 516–526, 2009.


**Ahmed Al-Jawad** is a PhD Researcher at Middlesex University, London, UK. He received his M.Sc. from the institute of Applied Computer Science, University of Freiburg, Germany, in 2008. He worked in several research institutes, where he published papers in the international conferences. His main research interests are related to the software defined networks, where he focused on the design of QoS management and network management, routing algorithms and video delivery.

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

**Purav Shah** is a Senior Lecturer in the Faculty of Science and Technology at Middlesex Univ., London. He received his Ph.D. from Univ. of Plymouth, UK, in 2008. His research interests include performance evaluation of wireless networks (protocols, routing, and energy efficiency), Internet of Things (IoT), UAV-based communication solutions and heterogeneous wireless networks. He is an IEEE member (MIEEE) and a reviewer of IEEE TCSVT, IEEE Access, Elsevier JSS, and has also actively served as TPC on several highly ranked IEEE conferences.

**Orhan Gemikonakli** is an Honorary Professor in telecommunications at Middlesex University, UK. He received his Ph.D. degree from the King's college London, UK, and he has been visiting professor at the University of Camerino, Italy. His current research interests include computer networks, wireless communication systems, Internet of Things, wireless sensor networks, parallel computing paradigms, integrated circuits, and performance/reliability modelling. He serves in various Programme Committees, and he is a reviewer for various journals. He is a member of ACM.



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