Policy-Based Network Management with End-to-End QoS Solution in Software Defined Networking

A thesis submitted to Middlesex University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Ahmed Nihad Abdulmajeed Al-Jawad
M00511788

Faculty of Science and Technology
Middlesex University London

October 2020

Abstract

Over the years the demand for large traffic volume and new advancements in network technologies have grown rapidly. The classical network in general has evolved towards complex architectures with vendor-specific designed interfaces. Unlike previous mobile network generations, 5G network provides a new foundational architecture with stringent requirements of network function virtualisation, massive scalability and high reliability with added flexibility. Alongside, the new concept of SDN-based network has offered potentially diverse benefits compared to the classical networks. Examples of the primary advantages include the centralized network provisioning and the network programmability. Therefore, Software Defined Network (SDN) with Network Functions Virtualization (NFV) have become a promising technique for the advanced 5th generation networks and the key components in the design of next generation networks.

As the Internet network is growing fast and operate wide range of traffic classes, the service quality and customer satisfaction are gaining importance in the network management. To this extend, policy-based network management becomes essential as certain traffic flows need to satisfy the business needs. In this research, a policy-based network management for quality services over SDN network is proposed. This research focuses on the traffic routing and measurement collection in order to satisfy the high level constraints. As part of the overall solution, the integration of intelligence in the framework is proposed. Machine Learning (ML) is becoming a very promising techniques to leverage the needs of autonomic and intelligent network management. This research proposes the use of Reinforcement Learning (RL) to enhance the decision making of policy-based network management for the end-to-end Quality of Service (QoS) guarantee. In this way, the proposed framework learns over time and it determines the best action to perform for ensuring end-to-end QoS delivery.

Recently the research on SDN has gained significant attention in the academia. In this context, the contributions of this research introduce novel methods to improve the monitoring and control towards intelligent traffic management solutions in multimedia-aware SDN-based environments. This research work brings three main contributions: (1) *measurement collection and probabilistic-based routing solution* is proposed to reduce the monitoring overhead over the control link between the forwarding and control layers in SDN, while increasing the observability of network state. This contribution applies a novel method based on sparsity approximation to compress the aggregated data in the SDN switch, while the SDN controller recovers the sparse data. Moreover, this research introduces an innovative probabilistic routing. The primary novelty of this contribution is the prediction of link bandwidth availability based on the Bayes' theorem. In contrast to other studies, the proposed routing algorithm calculates the routing path when less information is advertised by the switch plane; (2) *policy-based network management (PBNM) over SDN* to enable QoS provisioning. It addresses the use of SDN centralized architecture to adopt the standard policy-based network management, while retaining the proprietary modular-block at each layer independently. With the help of policy management, the SDN controller can meet the requirement of an end-to-end service delivery with QoS guarantee; (3) *reinforcement learning-based decision making for routing algorithms over SDN* is proposed to apply a novel approach based on reinforcement learning method for the dynamic routing algorithm selection under SDN-based environments. Based on the learning approach, the proposed solution selects the most appropriate routing algorithm from a set of centralized routing algorithms that maintain the flow satisfaction with respect to the defined SLA requirements.

The proposed solutions were evaluated under diverse scenarios. In order to evaluate the applicability of the overall proposed system, several tools are used for the experiments: MATLAB, Mininet (SDN network emulator), Floodlight SDN Controller, OpenVSwitch/Ofsoftswitch13 (Software switch). While, the following tools were used for the traffic generation: VLC (Live video streaming), Ostinato (Network traffic generator tool), FFMPEG (record, convert and stream video). For the evaluation, the following experimental setup (Linux-based machine) was used: SDN controller and application: 2.2GHz multiprocessor of 4CPU unit, memory size of 16GB, Mininet network emulator: 2.2GHz of 4 CPU units, memory size of 32GB. The research presents the design and implementation of the framework that leverages the benefits of SDN and performance evaluation results are discussed to validate the feasibility of this approach.

Keywords: Software-Defined Networking, Policy-based Network Management, QoS Provisioning, Monitoring, Artificial Intelligence, Machine Learning, Performance Metric.

Acknowledgments

I would like to thank Prof. Orhan Gemikonakli for giving me the chance to carry my thesis in his department. My special thanks I would like to express to my supervisors Dr. Ramona Trestian and Dr. Purav Shah for their consistent support and advice throughout this journey. My thanks also go to Dr. Ioan-Sorin Comsa for his guidance and for many fruitful discussions. Their valuable suggestions and guidance encourage me to explore deeply in the scientific material and to meet the research target. I would like to acknowledge my gratitude to my family members who supported me with no complaining at this tough period. Finally, I would like to express my very special thanks to my parents, who drove me to accomplish this long journey with motivation and encouragement. My parents supported me always during my pursue of the doctoral degree and they showed me that the doctoral degree is a long and tough journey, like a sailing ship, with full of excitements and potential pressure. To my dear mother and to my beloved father who died, I missed you and thank you for the support and passion. This thesis is dedicated for you.

List of Publications

1. **Ahmed Al-Jawad**, Ramona Trestian, Purav Shah, and Orhan Gemikonakli, "BaProbSDN: A probabilistic-based QoS routing mechanism for Software Defined Networks," IEEE Conference on Network Softwarization (NetSoft), 2015.

2. **Ahmed Al-Jawad**, Purav Shah, Orhan Gemikonakli, and Ramona Trestian, "Compression-based technique for SDN using sparse-representation dictionary," IEEE/IFIP Network Operations and Management Symposium (NOMS), 2016.

3. **Ahmed Al-Jawad**, Purav Shah, Orhan Gemikonakli, and Ramona Trestian, "LearnQoS: A Learning Approach for Optimizing QoS over Multimedia-based SDNs," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2018.

4. **Ahmed Al-Jawad**, Purav Shah, Orhan Gemikonakli, and Ramona Trestian, "Policy-based QoS Management Framework for Software-Defined Networks," IEEE International Symposium on Networks, Computers and Communications (ISNCC), 2018.

5. **Ahmed Al-Jawad**, Purav Shah, Orhan Gemikonakli, Ioan-Sorin Comsa, and Ramona Trestian, "Performance evaluation of routing strategies over multimedia-based SDNs under realistic environments," IEEE Conference on Network Softwarization (NetSoft), 2020.

6. **Ahmed Al-Jawad**, Ioan-Sorin Comsa, Purav Shah, Orhan Gemikonakli, and Ramona Trestian, "An Innovative Reinforcement Learning-based Framework for Quality of Service Provisioning over Multimedia-based SDN Environments," IEEE Transactions on Broadcasting, 2021.

7. **Ahmed Al-Jawad**, Ioan-Sorin Comsa, Purav Shah, Orhan Gemikonakli, and Ramona Trestian, "REDO: a reinforcement learning-based dynamic routing algorithm selection method for SDN," IEEE Conference on Network Functions Virtualization and Software-Defined Networking (NFV-SDN), 2021.

Contents

List of Figures

List of Tables

List of Abbreviations

ACI Application Centric Infrastructure

AI Artificial Intelligence

ARIMA Auto-Regressive Integrated Moving Average

CAIDA Center for Applied Internet Data Analysis

CC Cross Correlation

CR Compression Ratio

HD High-Definition

HDT Hold-Down Timer

IETF Internet Engineering Task Force

ISP Internet Service Provider

MHA Minimum Hop Algorithm

MIRA Minimum Interference Routing Algorithm

ML Machine Learning

MOS Mean Opinion Score

MP-BCP Most Probable Bandwidth Constrained Path

NFV Network Functions Virtualization

NMAE Normalized Mean Absolute Error

OMP Orthogonal Matching Pursuit

ONF Open Networking Foundation

OSPF Open Shortest Path First

PBNM Policy-based Network Management

PDP Policy Decision Point

PEP Policy Enforcement Point

PIP Policy Information Point

PSNR Peak Signal-to-Noise Ratio

QoS Quality of Service

RL Reinforcement Learning

RSVP Resource ReSerVation Protocol

SD Standard-Definition

SDN Software Defined Network

SLA Service Level Agreement

SLO Service Level Objective

SLS Service Level Specification

SNR Signal-to-Noise Ratio

SSIM Structural Similarity

SWP Shortest Widest Path

VBR Variable-Bit-Rate

WS Window Size

WSP Widest Shortest Path

Chapter 1

Introduction

1.1 Research Motivation

Looking back at the evolution of networking environments, the traditional networks have developed towards a complex architectures with custom designed interfaces. Due to the vendor-specific design, the vendors deploy network equipments for managing the entire control and forwarding of data packets in one physical unit. By this, the network devices are manufactured towards their own vendor-specific approaches and the network configuration has become a complicated process. As a consequence, network administrators require to gain an extensive knowledge of the individual tools and architectures [5–7]. With SDN, features in terms of management and programmability are leveraged to provide flexibility when compared to the traditional networks. SDN decouples the control plane from the data plane, while it offers a standardized interface between the planes defined in a communication protocol called OpenFlow [8, 9]. With the help of network abstraction, the new network paradigm shall centralize the network intelligence in the logical central unit. This add-on intelligence opens more opportunities to enhance the network services. In the future, 5G architecture is emerged with promising in flexibility improvement and separation between the control and user plane. This in turn gives the SDN concept key role to play on the future network architecture. The SDN-based network represents a new area for network performance improvement, that lately has attracted both academia and industry communities to investigate further challenges for an optimized network performance.

On the other hand, the QoS provisioning has become inevitably an important resource management issue in the enterprise networks. Network operators offer real time services while promising quality of service requirements for the voice and multimedia applications. QoS is defined in the ITU-T recommendation E.800 [10] as the collective effect of service performance, which determines the satisfaction degree of a user for a particular service. According to [11], the total IP traffic is expected to reach 396 exabytes of data per month by 2022, which is about 3 times higher than in 2017. Therefore it is necessary to prioritize the flows for QoS-based services. In order to do this, the network customers agree beforehand with the service providers in a formal contract described in a Service Level Agreement (SLA). The SLA represents a high-level policy of service specification which meets the business needs [12]. The SLA rules are mapped down to a set of low-level configuration rules on the network nodes. A policy-based management architecture is a solution to enable business rules to be translated and validated on a low-level configuration. In this research, the focus is on the traffic routing and measurement collection inside the policy-based management under SDN. The policy definition and translation are beyond the scope of the research objective.

SDN is relatively a new research domain and it is considered a very promising area for the research community [6]. Alongside, QoS provisioning is increasingly demanded by the industry to meet the business level. Together, SDN and guaranteed QoS provisioning are considered a motivation for further investigation towards a research aspect. This research has an academic impact in the networking area. It sets up a further important step forward to address the integration of Artificial Intelligence (AI) and computer networking. The results of this research study shall draw benefits to the academia and industry communities. This shall clearly spin-off the development of intelligent networking systems that rely on the integration of AI technologies in order to provide new functional opportunities.

1.2 Problem Statement

The Internet Engineering Task Force (IETF) has described a general framework architecture for Policy-based Network Management (PBNM) [13]. The PBNM framework presents two main architectural elements: Policy Decision Point (PDP) and Policy Enforcement Point (PEP) (See 2.3 for details). The PDP decides on the action based on the state of the PEP whether it fulfills the business needs. The policy information is stored in a repository Policy Information Point (PIP). Although the IETF model has described the functional entities, there have been efforts by research communities to design the functionality with decision-making algorithms [14]. It becomes also a major challenge to obtain a self-configuring system that complies with the desirable requirements. Literature has shown an area of concern within the decision unit (PDP) inside the policy-based network framework [15–17]. To this extend, this study presents the utilization of intelligent decision-making solution for the proposed framework. Hereby, this research utilizes the ML techniques to apply an adaptive bandwidth provision scheme based on the learning from the experiments of past trails. With the help of ML, the network management for end-to-end QoS becomes self-configured and self-optimized. The utilization of routing management and data collection is studied in order to adapt dynamically the network behavior by the defined policies in PBNM. Here are the research problems that are addressed in this thesis:

- **Static Network Configuration:** The global network continues to grow significantly and the network technologies such as 5G architecture are advancing rapidly. This in turn, makes the static network configuration to become an impractical approach when coping with all the network and business needs [18]. In fact, providing an improved QoS level while utilizing the network resources efficiently is becoming challenging to the service providers. Nowadays the network is changing state over time due to the variable nature of traffic patterns, therefore it is very time consuming and failure-prone to enforce the static configurations on each individual network node.

- **Innovative QoS Routing:** The QoS route is defined as the travelling of data packets along a feasible path between the source and destination nodes which satisfies the constraints given by the QoS requirements. However, a scheme to provide a reliable end-to-end QoS is still one of the main remaining open issues and it becomes difficult to satisfy the stringent service level requirements that belong to different application classes over the network. For example, video calling and multimedia applications demand more stringent quality requirements than web browsing in terms of parameters like loss, delay, and bandwidth. This challenge arises when the resource availability in the network is limited, which leads to high levels of congestion and consequently poor quality. In the meanwhile, the significant growth of multimedia-based applications, makes it obvious that just an increase in the system capacity will not be enough to meet the QoS requirements of different application classes. By this, the routing algorithm needs to adapt to the dynamic changes in the network (e.g., the newly arrived traffic flows and the termination of existing traffic flows), the transient load fluctuations and the network limitations.

- **Low Service Quality Performance:** The impact of service quality performance on user satisfaction is a critical metric that needs to be improved with the help of an intelligent network management. While delivering network services and applications to users, the performance of quality services is an important measure to sustain a reliable user's quality of experience. By this, network services and applications require different levels of quality performance. For example, low latency and minimal jitter is required for applications such as live network streaming, while reliability is required for applications involving file transfers that cannot tolerate packet loss or high delay. Over time, the network can exhibit different levels of congestion and the user's quality of experience decreases depending on factors like the packet loss and latency. As a consequence, it is essential to design an advanced intelligent system to utilize the network resource more efficiently and more fairly.

These problems can lead unpredictably to bad network performance and poor resource utilization. In fact, all these problems may affect the network operators in terms of possible loss in revenue and customers churn rate. In this context, this research aims at presenting a novel approach to solve the above issues by proposing a learning approach for end-to-end QoS provisioning within a policy-based framework. Solving these problems remains an interest to the service providers due to the need in providing a promising outcome to the customers while at the same time reaching an efficient network resource utilization [19].

1.3 Solution Overview

This thesis presents a solution concept to deal with the problems as identified above in 1.2. The proposed framework employs statistical decision-making methods for fair traffic distribution while guaranteeing the QoS requirements. As a result, the approach reduces the risk of network congestion and effectively utilizes the available network resources. In order to do that, this research proposes a framework that integrates the use of AI, PBNM and SDN for the end-to-end QoS optimization (see Figure 1.1).

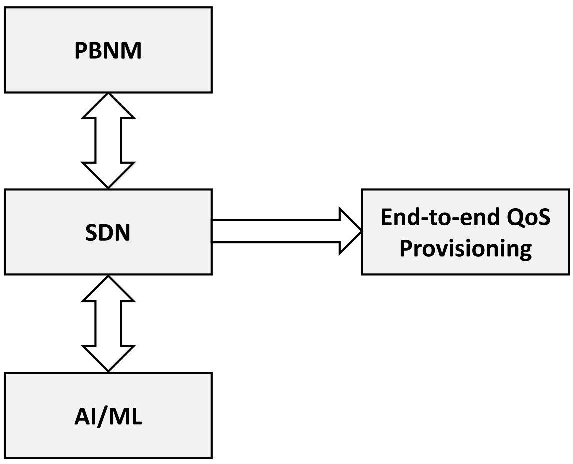

Figure 1.1: View of the applied techniques in the framework

As part of the solution, figure 1.2 shows the closed control loop in the network management. The figure presents the three main elements in the network management to maintain a monitored network complying to the QoS requirements. As the network is steadily monitored by the observation unit, the identification uses the measured quality metrics, accordingly, to scan the high-level policies whether a requirement is being violated. As a result of a violation, the decision-making unit needs to take the measure to resolve the artifacts. Because the network is usually incorporated with dynamic change, designing the internal entities inside the closed loop becomes a key challenge in achieving high quality performance. By introducing the closed-loop control structure (see Figure 1.2), a self-organization concept can be realized in the network.

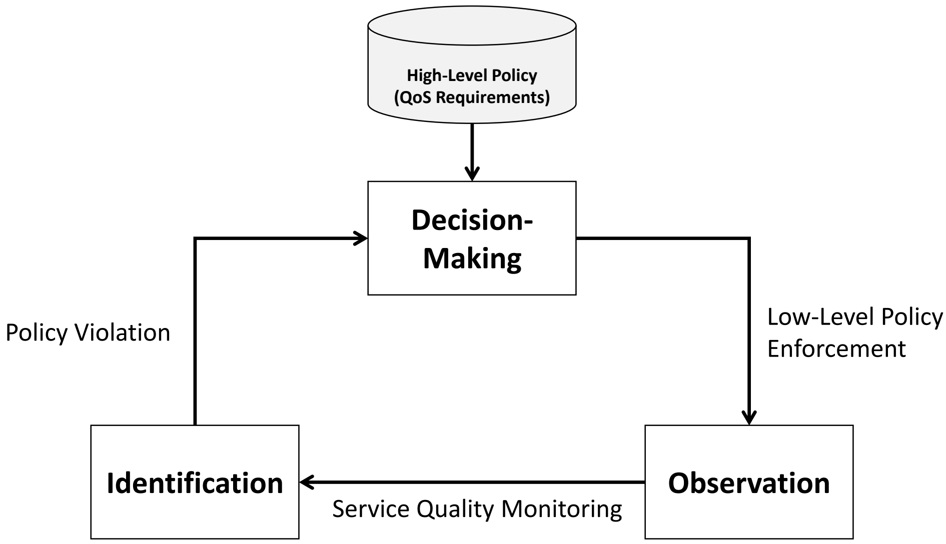

Figure 1.2: A general closed loop network management system

The proposed solution makes use of PBNM to facilitate the routing management over SDNs. In the field of AI, the RL becomes a promising solution for a centralized network management. In comparison to other supervising learning methods, the objective of RL is to optimize the network performance by finding the optimal action through maximizing the expected reward. In this thesis, a novel framework that employs RL in the decision-making unit of the PBNM system is proposed. Through learning, the system iteratively improves its knowledge every cycle iteration. The RL algorithm consists of three main elements: reward, action, and state [20]. The reward and state can be derived from the network state which is monitored steadily. While the action space contains the possible actions, for instance re-routing, that can be used to retain a certain policy requirement. By integrating RL in the PDP component, the intelligence reacts proactively on inadequate network changes by finding the optimal selection of action set to maximize the long term reward. In general, RL can be adopted to fit the closed-loop structure defined in figure 1.2.

On the other hand, the key features of SDN include the separation of the control from the data plane, the logical centralization of the network control, the ability of programming the network, and the visibility through maintaining a global image of the network by the SDN controller. By taking the advantages of these features, this research work shall draw the benefits of the SDN architecture to the proposed framework. Towards this end, a RL-based approach running over SDN to tackle the aforementioned challenges is proposed. The proposed solution, as depicted in figure 1.3, combines the advantages of the autonomic PBNM system and the SDN architecture for QoS management.

To this end, the overall goal of this research focuses on four pillars: QoS, SDN, PBNM and ML. The study presents at how the integration solutions between the four aspects are utilized in a single network management framework.

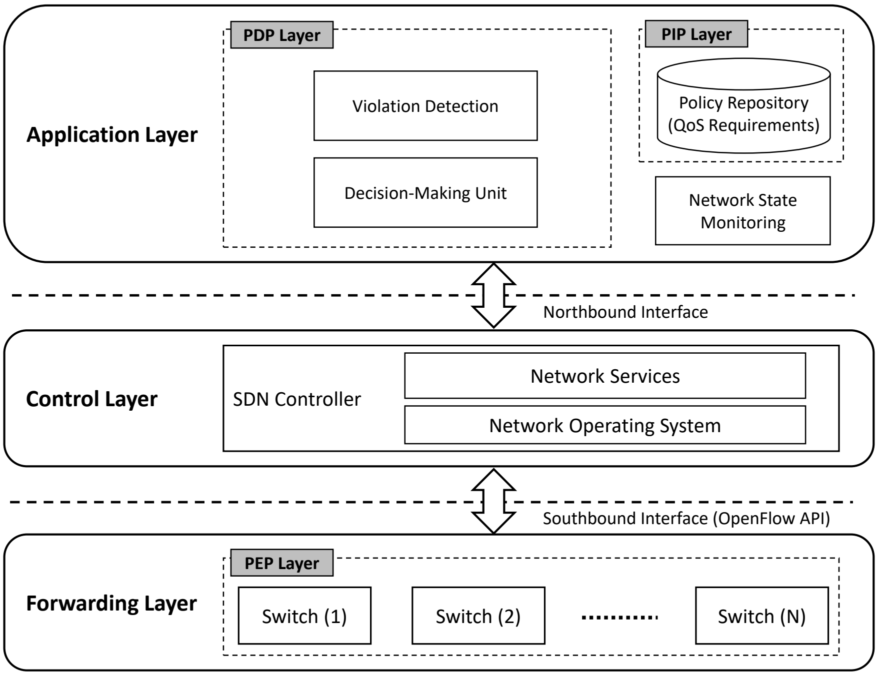

Figure 1.3: A general high-level overview of the proposed solution

1.4 Research Contributions

As the network is a highly dynamic environment in which, for example, the traffic patterns are changing over time, it is essential for a network monitoring solution as part of the network management to measure the quality performance of the underlying network layer. Though on the other hand, the network management based on routing mechanism represents an important function and capability to drive the network to ensure a high user expectation of network services. In this research, the objective aims to study the network framework for monitoring and management under the SDN architecture, while using a realistic experimental setup to evaluate the framework. The contribution investigates further the employment of an innovative approach for monitoring and management. In order to realize the research objective, the research work presents the following main contributions:

- Proposal of measurement collection and probabilistic-based QoS Routing for SDN: The proposed framework builds on the classical SDN architecture and adds two main contributions: compression-based monitoring solution and probabilistic routing solution. The compression-based monitoring solution aims at further decreasing the link usage for QoS applications while increasing the network observability. It differs from other traditional methods like polling the SDN switches' statistical data. This innovative approach makes use of the sparsity approximation to compress the aggregated QoS metric data. It extends the functionality of the SDN switch by integrating dictionary learning algorithms like K-SVD and Orthogonal Matching Pursuit (OMP) methods for the purpose of sparsity approximation. Additionally the contribution integrates the probabilistic scheme for the QoS routing algorithm that employs the bandwidth availability metric as a QoS routing constraint for data delivery. It makes use of Bayes' theorem and Bayesian network to determine the link probability in order to select the route that satisfies the given bandwidth constraint. It considers the amount of overhead while evaluating the QoS performance in terms of bandwidth blocking rate. The introduction of link state update policies reduces the amount of state information packets exchanged between the control and data planes. The proposed solution of this research contribution and the results of the study are presented and discussed in [21, 22].

- Proposal of Policy-based QoS Management Framework for Software-Defined Networks: This contribution presents the integration of the policy based framework and the SDN from the architectural point of view. It presents a proof-of-concept for the use of hierarchical structure of the standard policy-based network management on the centralized management plane of SDN architecture, while being able to keep the set of functional blocks (e.g. route manager, violation detector, etc.) at each layer independent of the policy management architecture. By this, the proposed framework monitors the QoS parameters and enforces dynamically new decisions (such as: rerouting or rate limiting) on the underlying SDN switches to ensure the demand of the high-level policies. The introduction of feedback closed-loop between the underlying SDN switches layer and the application layer brings better performance to the evaluation results when compared to the default SDN-based network. The proposed solution has been even extended further in the upcoming contribution by introducing an intelligent decision making unit. The proposed solution of this research contribution and the results of the study are presented and discussed in [23].

- Proposal of a Reinforcement Learning Approach for enabling QoS over PBNM-based SDN networks: The contribution introduces an innovative Reinforcement Learning-based framework for multimedia-based SDN environments. The proposed RL-based framework utilizes the Q-learning method to decide on the most suitable routing algorithm to be applied on the traffic flows in order to ensure the QoS provisioning. The aim is to select dynamically the most appropriate routing algorithm from a set of centralized routing algorithms that maximizes the return reward from the network. The return reward indicates the network performance improvements based on the QoS parameters. The proposed RL-based solution was implemented and evaluated using an experimental setup under a realistic multimedia-based SDN environment. The proposed solution of this research contribution and the results of the study are presented and discussed in [24–26].

This research elaborates solutions towards the network monitoring and controlling of the traffic flows in SDN-based environment. The network management monitors the underlying network layer and it controls the traffic accordingly to meet the user satisfaction needs. In this research, the first contribution studies the monitoring scheme under the SDN-based and it proposes a novel method based on the sparsity approximation to compress the statistical data to the controller, while it also discusses a new approach of routing based on probability of the update information. The following contribution evolves towards the integration of monitoring and controlling under the policy-based management and SDN architecture. It introduces the feedback closed-loop between the underlying SDN switches layer and the application layer and it highlights the benefits in terms of better network performance. This leads to the final contribution by integrating the learning based scheme to control the traffic intelligently and choosing the appropriate routing algorithm that fits the network state.

1.5 Thesis Structure

This thesis is structured as follows. Chapter 2 gives a general introduction to the technical concepts that are utilized in the work. Chapter 3 presents a review of related work in the research area. It illustrates a survey of the state-of-the-art literature in the field. In chapter 4, the overall proposed system architecture and operational definitions will be presented. This chapter describes the main contributions in the research. In the following chapter, Chapter 5, presents a data collection scheme based on a compression technique for SDN-based network. Additionally, the chapter shows the probabilistic-based QoS routing mechanism for SDN. The approach uses Bayes theorem and Bayesian network to determine link probability. In Chapter 6, the policy-based management framework to enable QoS provisioning over SDN-based networks is presented. By using loop chain approach between network monitoring and policy validation/enforcement, the framework can achieve end-to-end QoS guarantee. Chapter 7 presents a comprehensive performance evaluation of the state-of-the-art routing algorithms under realistic scenarios with dynamic network conditions and various topology. While chapter 8 shows the integration of reinforcement learning algorithm for routing management under SDN environment. The closing chapter, Chapter 9, contains a summary of the study and suggestions for future work.

Chapter 2

Technical Background

This chapter introduces the technical background for the work presented in this thesis. It is structured into five main areas, these are: Quality of Service, Software Defined Networks, Policy-based Network Management, Routing Algorithms and Artificial Intelligence.

2.1 Quality of Service

Nowadays diverse traffic classes (such as video and data) are transferred across the communication networks. Due to the limitation of network resources, the quality of the traffic flows is affected as a consequence from the network congestion. This leads to several traffic problems such as packet loss, low throughput, high jitter and delay, etc. which have a great impact on the users' perceived quality. Therefore, to guarantee certain requirements, traffic classes possess different QoS requirements. Guarantee of QoS provisioning has become an active field of research especially considering its importance for applications that require data delivery under certain QoS constraints (e.g., multimedia and voice data). The QoS-enabled networks provide significant performance improvements for QoS services by ensuring sufficient bandwidth, controlling latency and reducing packet loss.

For evaluating the performance of QoS-based applications, there are some important metrics to be considered such as: throughput, packet loss, and delay. Network performance metrics have different meaning in different aspects, hence they impose different measurement methods. The definitions of some of the important QoS performance metrics are given below:

- **Throughput:** It is defined as the amount of data being sent or received per time unit. Another important related metric is the available bandwidth, though it has another meaning. The throughput means the measure of actual packet delivery over the link, while the bandwidth availability indicates the amount of link capacity is available to allow additional traffic flows to transport over the link. Based on the throughput measurement, the available bandwidth of a link can be induced.

- **Packet Loss:** It is defined as the number of lost packets on the path between the sender and receiver across the network. In general, there are many causes of packet loss such as the network congestion, finite buffering capacity or errors in data transmission. The packet loss rate can be calculated as follows [27]:

$$PLR(\%) = \frac{n^{tx} - n^{rx}}{n^{tx}} \times 100 \tag{2.1}$$

  where n^{tx} and n^{rx} are the total number of transmitted and received packets, respectively.

- **Delay (Latency):** It indicates how long it takes for a data packet to travel from the source to the destination. The total delay between two network nodes is usually the sum of several parts like processing delay, queuing delay, transmission delay and propagation delay. The main cause for packet delay is network congestion.

In general, QoS is attributed with a wide range of performance metrics such as packet loss and throughput. A path metric is the composite of same metric through all the links along the path. In fact, the composition way depends on the metric type and there are three basic kinds of metric composition rule [28], such as:

- **Additive Metric:** The value of QoS parameter constraint $m\left(p\right)$ is the sum of a single metric' values $m\left(L_i\right)$ along the path p. Examples of metrics which obey the additive composition rule are delay, jitter, hop count and cost. The rule is represented as follows:

$$m\left(p\right) = \sum_{i}^{n} m\left(L_i\right) \tag{2.2}$$

- **Multiplicative Metric:** The metric multiplication of all links along a path represents the overall path metric. The reliability and loss probability are examples for this composition rule. The rule is described as follows:

$$m\left(p\right) = \prod_{i}^{n} m\left(L_i\right) \tag{2.3}$$

- **Concave Metric:** Here the maximum or minimum value along all the links of a path is selected for the value of QoS parameter constraint. Example of this metric is bandwidth availability:

$$m(p) = min/max\lfloor m(L_i)\rfloor \tag{2.4}$$

2.1.1 QoS Provisioning for Multimedia Delivery

Nowadays considering QoS requirements in network communication is becoming very essential to support reliable service for end users. According to Cisco forecast, video traffic volume will reach 82% of all IP traffic by the year 2022 [29]. Therefore it is important to consider the requirement of QoS for multimedia applications in order to guarantee a sustainable service delivery to the users. Multimedia has been applied to wide applications such as one-way video streaming and interactive video conferencing. In this thesis, the focus is on video streaming. The main QoS performance metrics that impact the user perceived quality are throughout, delay, and packet loss. For one-way video streaming there is no real-time interaction involved, therefore it can tolerate certain threshold levels of delay and packet loss rate [30].

2.1.2 Approaches for Measuring the Video Quality

The video quality evaluation has been extensively studied to describe the quality level of video. In general, there are two techniques to assess the video quality: subjective and objective [31]. The subjective quality evaluation is based on the individual human perception for rating video quality. In subjective assessment, a number of viewers are asked to rate the video quality and the average rating over all viewers is called Mean Opinion Score (MOS). In the meanwhile, there is a number of techniques that is employed to evaluate the video quality subjectively [32].

On the other hand, the objective quality evaluation is based on algorithms for characterizing the video quality. Different types of objective evaluation exist, they can be classified into full reference, no reference and reduced reference metrics based on the amount of reference data [33]. In full reference method, the original image is used a reference information and the received image is compared against the reference. In the reduced reference, reduced information are expressed in number of features like texture that are extracted from the original and received images. The reduced information are compared between the two images. On the other hand, no reference method required no data from the original image. However assumption about the original content is used under this method. By this, the video or images are analyzed without given reference data [31].

The most commonly used objective metrics to assess the video quality is Peak Signal-to-Noise Ratio (PSNR). The PSNR is an objective video quality assessment method that makes use of the reference image to compute the ratio between the maximum possible value of the image and the error power [34]. A higher PSNR indicates better image quality, by this if the compared images are identical the PSNR becomes infinity. It is expressed in logarithmic decibel dB. The PSNR of two images x and y is computed using the following equation:

$$PSNR\,(x, y) = 20 \cdot log_{10} \left(\frac{MAX}{\sqrt{MSE}} \right) [dB] \tag{2.5}$$

where MAX indicates the maximum possible value of the video frame pixel and MSE is the mean squared error between the original frame x and received frame y of width n and height m. It is calculated as follows:

$$MSE\,(x, y) = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [x\,(i, j) - y\,(i, j)]^2 \tag{2.6}$$

Since the video is comprised of multiple frames N, the PSNR of a video can be calculated by comparing the PSNR with the original video on a frame basis. The overall PSNR is the average of all PSNR of N frames.

Another objective video quality assessment method is the Structural Similarity (SSIM) index [35]. SSIM is based on the structural information of the video frame. The index range is measured between -1 and 1 where 1 represents the maximum video quality as the video frames are identical, while -1 indicates maximum difference. SSIM combines three factors, such as: luminance, contrast, and structure and is given by the equation below:

$$SSIM\,(x, y) = l\,(x, y) \cdot c\,(x, y) \cdot s\,(x, y) \tag{2.7}$$

where $l\,(x, y)$ is the luminance distortion between the two video frames, $c\,(x, y)$ indicates the contrast comparison and $s\,(x, y)$ measures the correlation coefficient between two video frames.

Literature like [1] introduces a mapping of PSNR and SSIM to the nominal MOS as given in Table 2.1. MOS represents a five point scale, with 1 being bad and 5 being very good. It is used to subjectively rate the users' perception of video quality.

Table 2.1: PSNR and SSIM to MOS Mapping [1]

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

2.2 Software-Defined Networking

Nowadays, the global network is evolved towards complex architectures and custom designed interfaces, where the classical routers have their own responsibility to perform the control, management and forwarding of data packets. In contrast, SDN has emerged with a wide spectrum of benefits [36]. For instance the centralization, programmability via standardized interface, the decoupling of control and forwarding planes, and feasibility through global network image. All these features can simplify the management, network interface, and programming compared to the traditional networks. The new network concept is simplified by decoupling the control plane from the forwarding (data) plane. Figure 2.1 shows the architecture view of SDN-based network. The SDN paradigm consists of three layers: application, control and forwarding plane. The data layer is realized by SDN switches and it is responsible for forwarding the data packets based on the configuration from the control plane. While the SDN controller is located in the control layer. The controller configures the network switches by publishing the rules to the switches. In order to do this, the communication between the two layers is achieved through the OpenFlow protocol via a southbound interface. Along the southbound interface, there is an open northbound interface represented in a user-defined API. The interface is located between application and control layers. The network applications reside in the application layer and they communicate with the SDN controller via this interface [37]. Example of network applications is the QoS and security network management.

The SDN switches maintain a group of flow-tables. Each flow-table contains a set of match-action rules. The flow entry describes the action to be taken when the data packet matches the rule. The flow-table of switches are configured by the controller via defined libraries in network services. Additionally, the flow table in SDN switches maintains statistical information per flow. Therefore the forwarding plane provides a direct measurement that belongs to the flow-level monitoring. In general, an entry in a flow-table has four fields: rule field which consists of various information such as source and destination IP or MAC addresses, action field (forward, redirect and drop) that defines how the packet will be processed, priority field in case a packet has more than one match and lastly statistic field for management uses [38]. Generally, the control plane in SDN architecture can contain several independent controllers, but in this research a paradigm with a single controller will be discussed.

2.3 Policy-based Network Management

The static network settings are based on configuring the network devices individually. Usually the policy rules are restricted to static condition-action format, where a certain action is applied when

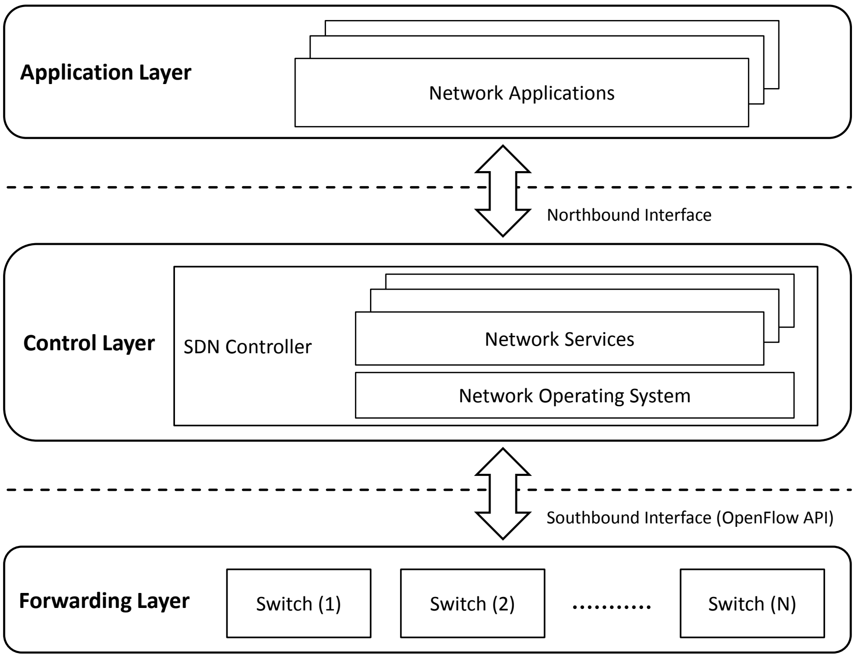

Figure 2.1: The SDN architecture

the condition is met. For upcoming changes in network configuration, manual intervention by administrators becomes necessary. In the meanwhile, the main objective of policy management is to ease the burden of network management on the administrators. As the network state is rapidly changing due to the traffic dynamics and complex structure, it makes manual management by network administrators rather impossible to configure over time. PBNM has become a promising solution for network management, especially for large scale networks where, changes to accommodate applications' requirements occur dynamically. The management system has the advantage of altering the system configuration dynamically according to the network behavior and settings. Generally, PBNM systems can be used for several applications such as security and resource management. However, in this thesis the focus is on the use of PBNM and QoS.

In the network operation life-cycle, PBNM enforces the policies on the underlying network devices and adaptively validates the network behavior against the policy context. As a result of policy violation, the decision-making unit should choose an adequate action to reduce the risk of a foreseen violation in the upcoming time window. In terms of QoS-based data traffic, a violation means the network does not fulfills the QoS requirements of policy [39, 40]. As a result, it becomes very challenging to select the best action among a set of possible measures in order to retain a maximum long-term reward in terms of network performance. Hence, this kind of problem is of focus to be further investigated by the current research.

In the context of QoS, the PBNM controls the resource provisioning in order to meet the business requirements. The agreement consists of a set of high-level policy rules that govern the decision on data layer. In order to offer a consistent end-to-end QoS guarantees, the Internet Service Providers initially sign a SLA with the enterprise customers. The SLA represents a non-technical high-level service specification and it defines the demand conditions for guaranteed services at the business level. The next interpretation stage of SLA is the Service Level Specification (SLS). The SLS is derived directly from SLA and it contains the Service Level Objective (SLO) as sub-item that describes the technical interpretation with measurable terms (e.g. throughput, packet loss, delay, etc.). In this thesis, the definition and translation from SLA to SLO are beyond the scope of the research objective. Here, the research utilizes the routing management and data collection in order to adapt dynamically the network behavior by the defined policies.

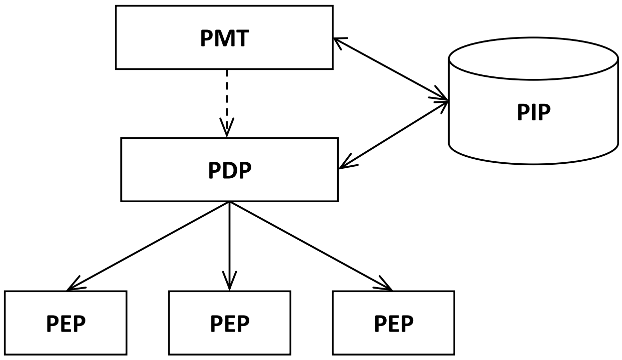

Figure 2.2: Traditional PBNM architecture

Figure 2.2 illustrates a standard architecture for PBNM as defined by [41]. Four entities are defined: **(1) Policy Information Point (PIP)** - contains the policy information and requirements. The policy is defined as the combination of rules and services where rules define the criteria for resource access and usage [13]. In this research, the focus is on the QoS requirements in the policy. Other entities in PBNM can store and retrieve policy information to and from PIP; **(2) Policy Decision Point (PDP)** - makes decisions based on policies and requirements stored in PIP. It is responsible for handling network events, making decisions and communicate them to PEP; **(3) Policy Enforcement Point (PEP)** - dynamically enforces network low-level policy rules on network elements, such as switches, as directed by PDP; **(4) Policy Management Tool (PMT)** - software tool for network administrators to manage and edit the policy rules.

2.4 Routing Algorithms

In general, the traditional network architectures suffer from several main limitations such as the lack of global view, per-hop decision. This is mainly due to the fact that control is distributed and implemented on network devices [42]. With the introduction of the OpenFlow-enabled SDN network, the routing algorithms benefits from the new concept. Through the centralization and programmability, flow-based routing becomes feasible and scalable. By this, OpenFlow provides more flexibility of defining a set of actions that are associated to various types of flows. This enables each flow to be treated differently in the underlying network devices. Furthermore, centralized monitoring and routing can help network operators to create powerful management framework under SDN network [42,43]. By definition, the routing algorithm is used to find a feasible path from a given source node to a given destination node that satisfies a set of constraints while achieving efficient resource utilization. A path is considered feasible if it satisfies the given requirement, for instance if the available bandwidth capacity is higher than the requested bandwidth. In general, there are different parameters that characterize the network performance such as the bandwidth, delay and loss probability. However, when multiple network constraints are used in the routing determination, the problem of finding a path becomes difficult to solve. Solving the problem of multiple constrained QoS routing is NP-hard [28, 44]. In this research, the routing algorithms for the bandwidth constraint are addressed. On the other hand, this research shows the importance of machine learning and SDN architecture in addressing the problem of centralized routing and the selection of an appropriate routing algorithm. For this research in order to enable QoS provisioning within a network, the resource availability of the network must be monitored continuously.

To describe the routing algorithm, a network topology is modeled by a graph $G(N, L)$ where N is the set of nodes and L is the set of links. In the network, each link has a fixed capacity. As various traffic flows pass through the link, the available bandwidth is the residual capacity of a link that remains after the cross traffic is served. When a path setup request arrives, a reduced network graph is created by eliminating all links that have residual bandwidth less than the requested one. Based on the reduced network graph, the routing algorithm finds a feasible path where the links have residual bandwidth equal or greater than the demanded bandwidth. In this research, four centralized routing algorithms for traffic requiring bandwidth guarantees are presented:

2.4.1 Minimum Hop Algorithm

The Minimum Hop Algorithm (MHA) chooses the path with the minimum number of links between the source and destination nodes [45]. The algorithm is based on Dijkstra's algorithm in finding the shortest path, where the cost of a link is set to 1.

2.4.2 Shortest Widest Path Algorithm

The Shortest Widest Path (SWP) Algorithm finds the feasible path with the maximum available bandwidth among the set of existing routes [45]. If there are multiple such paths, the one with the shortest route is selected. However, if there are many paths with the same bandwidth availability and shortest route, then randomly one is chosen. The Dijkstra algorithm is used to select the feasible path, where the cost of a link is assigned to the residual bandwidth. In general, the links that cannot satisfy the bandwidth requirement are removed before applying Dijkstra algorithm.

2.4.3 Widest Shortest Path Algorithm

The Widest Shortest Path (WSP) Algorithm finds the feasible path with the shortest path among the set of existing routes [45]. If there are multiple such paths, the one with the maximum available bandwidth is selected. However, if there are many paths with the same bandwidth availability and shortest route, then one is randomly chosen. The algorithm tries to balance the network load and it relies on load balancing to avoid network bottlenecks.

2.4.4 Minimum Interference Routing Algorithm

The 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 is arriving [46]. In order to satisfy future demand, the algorithm attempts as much as possible to avoid placing the route requests along the critical links. The critical links are the links that decrease the max-flow of other ingress-egress pair if they are selected in the path. The interference is computed based on the maxflow-mincut theory, where Links that belong to a mincut set are considered critical links [47]. In general, the algorithm goal is to find the least critical feasible path.

2.5 Artificial Intelligence

AI is a branch of computer science, that helps machines perform cognitive tasks based on an intelligent interpretation of the ambient conditions in order to achieve some defined goals. In recent years, AI has become very popular in the research and the technology industry. It has been applied intensively to many fields like the robotics and visual perception [48]. AI has diverse techniques that are equipped in machines to make them behave and decide in an intelligent manner. Knowledge engineering is considered the main part of AI where gathered data from the environment is analyzed to extract adequate information. In general, there is no unified and unique AI paradigm to be applied for solving the existing problems in different fields. AI includes diverse methods and they need to be structured together to fulfill the application needs.

Another important core part of AI is ML. It enables computer systems to learn from experiences. Learning takes place by generating and collecting training data. The approach becomes very promising for decision-making system. It has been applied to wide spectrum of applications like finance and computer vision. ML has mainly three approaches: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. In this research, the focus is on RL due to the nature of this technique that fits the research problem. RL is used to learn the agent to choose the best action by trial-and-error. By this, the agent interacts with an unknown dynamic environment and by using the feedback it learns and decides the action to perform [20]. The RL algorithm consists of three main elements: state space, action space, and reward. Unlike other machine learning methods, Figure 2.3 represents the RL agents that learn from the interaction with the environment. Initially, the agent observes the state of the environment and chooses the action that maximizes the expected long-term reward. Based on the agent interaction, the environment returns a numerical reward accordingly, which the agent aims to maximize. However, there is a trade-off between exploration and exploitation in RL. 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 actions may perform better.

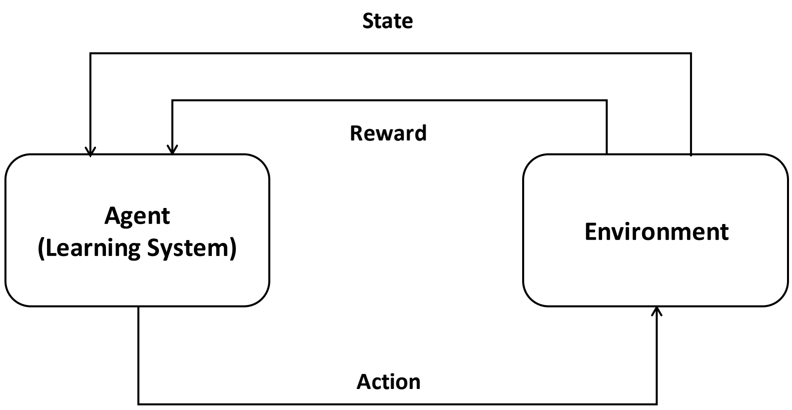

Figure 2.3: RL, Agent, and Environment

2.6 Chapter Summary

This chapter presented the background technologies used for this research work. QoS delivery represents a vital feature in networking nowadays especialy with the increase of multimedia-based applications which are known to have strict QoS requirements. This work focuses on the delivery of QoS over SDN-based environments for multimedia-based applications. The QoS provisioning for multimedia traffic and the main approaches for measuring the video quality as perceived by the end-user are discussed. Moreover, the new networking paradigm, SDN was introduced. It has been identified that SDN has key features that can bring major benefits to the network. Additionally, the basis behind policy management systems and the functions of its components were described. Finally, AI could play an important role for enhancing the system performance. Thus, one of the main goals of this research is to integrate the above mentioned technologies into a single intelligent network management framework.

Chapter 3

State-of-the-Art

This chapter presents a comprehensive survey of the current research on the following topics: QoS provisioning over SDN, PBNM for SDN, routing management and the use of AI in SDN. The existing industry solutions, and solutions approaches in the research literature are presented and compared. The main challenges and open issues that need to be addressed in the evolution towards a unified solution for enabling QoS over multimedia-based SDN networks are outlined.

3.1 Current Industry Solutions

Nowadays, Software Defined Networking represents an attractive solution for the industry for driving the traditional network solutions to the new era of programmable networking management. Due to this, a new nonprofit trade organization Open Networking Foundation (ONF) was founded by several well known networking companies to promote networking-related innovations towards SDN solutions [49]. The ONF origination defined the OpenFlow standards and SDN architecture [50]. Alongside, the requirements of advanced 5G network such as large degree of networking flexibility and high reliability gives the SDN concept more importance to be an enabler in the new 5G network. The main feature of SDN is the separation of the control plane from the network hardware, which places SDN as one of the main candidates for the achievement of 5G requirements [51, 52].

With respect to market trends, different SDN controller vendors like Brocade, Cisco, Juniper, or Aruba have brought the realization of SDN controller towards the commercial level. However, some networking companies like Brocade and Extreme Networks leverage on the OpenDaylight framework for their SDN controller solutions. Other controller like Floodlight was used widely for teaching and research [53]. The Floodlight controller is originally based on Beacon controller from Stanford university and it provides simpler programing interfaces with various code samples. On the other hand, several hardware OpenFlow enabled switches are introduced in the commercial market. For example: Hewlett Packard presented a high performance switch to be used within the SDN infrastructure. Network companies like Brocade, Extreme Networks, Dell and IBM came out as well with their own OpenFlow-based vendor-specific SDN switches [54, 55].

Moreover, many enterprises accelerated the innovative solution by offering the entire network infrastructure of SDN to the end customers. Such a solution supports wide examples of network services like data center, cloud applications, policy management, and networking visualization. Cisco, as one of the major player in the networking industry, presented the SDN concept through the introduction of Application Centric Infrastructure (ACI) and Cisco Nexus 9000 series switches [56,57]. The solution unveils for the entire SDN infrastructure and through partnership with IBM, offers wider networking solutions like the full life-cycle services. On the other hand, Nokia, Verizon, and Extreme Networks presented a business solution with SDN infrastructure. Large service providers like Google are deploying the SDN stack in their network infrastructure and they boost it with the networking services for SDN. Another acceleration towards SDN adoption, VMware has launched the network virtualization for the SDN [58–60].

Beyond that, the main unique objective which makes the SDN concept more popular among the industries is the sharing of common programmable interface. Despite the networking devices being manufactured with different hardware modules, they are kept to comply to the requirements of the OpenFlow interface. Hence, different devices by various vendors can be integrated easily by using the same OpenFlow interface [61].

3.2 QoS Provisioning in SDN

This section presents an overview of the existing state-of-the-art in SDN network management for QoS provisioning. Various research groups have studied different approaches to enable QoS provisioning over SDN-based networks. Earlier studies have investigated the role of routing algorithms in traditional network architecture [62, 63]. 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 and Openflow with the main focus on wireless networks is presented by Araniti et al. in [64]. The study in [65] presented a comparison of different centralized routing algorithms under the SDN network. The study considers two constraints for the routing calculation, namely the bandwidth and path delay. The bandwidth rejection ratio and the average route length are used to evaluate the performance of the routing algorithms. However, the performance evaluation was limited to specific network scenarios in terms of topology size and network loads.

MC Lee et al. [66] presented a heuristic algorithm for segment routing under SDN-based network. The solution considers the link residual bandwidth and it takes into account the current loading of each link while it estimates the predicted future traffic load on the link. The results show that the proposed method outperforms other routing algorithms in terms of average rejection rate and average network throughput. However the drawback by using the segment routing for routing determination is the requirement of extra per-packet header size overhead in order to represents the route information as a set of labels. The simulation setup is based on JAVA program to model different network environments and different traffic flow distributions. On the other hand, the work in [67] 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 in the research field is the flow routing framework [68–75]. Under this method, the flows' routes are adjusted dynamically according to the network state. This solution uses mainly the key features of SDN such as the utilization of the global network view to compute the proper route and the centralization of network management. One of the earliest contributions in this area is Egilmez et al. in [68–70]. The authors presented an optimization framework for QoS routing under SDN-enabled network. The work is based on an end-to-end dynamic QoS routing solution for the multimedia delivery over SDN-based networks. The work proposes and solves different optimization problems to reroute the base layer of video streams on alternate routes. In one of their study, the experimental setup is based on three hardware OpenFlow-enabled switches and three host computers that are connected to form a triangular network topology [68]. While other study by them, the simulation test was based on the Georgia Tech internetworks topology modeling tool. However it is unclear how the integration of OpenFlow-enabled network is taken place under this tool [69, 70].

Other works like [71–73] presented an approach for adaptive video streaming over SDN networks. The work in [71] treated the base layer and enhancement layer of video streaming separately on different routing paths. For the case 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. The experimental test was based on mininet and floodlight, the results show that packet loss rate of base layer packets can be improved when the load level of the shortest path increases. While the work in [72] routes all the flows in the network along the shortest path, with the minimum cost. The work solves the the optimization problem that is formulated as integer linear programming for the multi-commodity flow and constrained shortest path model. The scenario is composed of both an emulated network using mininet network emulator with a real wired and wireless network topology. By this, the topology is composed of six switches and various servers like video straining server to emulate the traffic for the experiment. The results shows that it is possible to avoid the link congestion effects and the solution finds a mapping between the mathematical basics and the quality assessed by the user. The work in [73] proposes a network path selection scheme by rerouting the flows based on the predicted measurements. The work employs Kalman filter to predict the bit rate in the upcoming time frame. By this, the solution reroutes the flows based on the predicted measurements. The experiment setup is composed of four OpenvSwitches switches, OpenDaylight controller and media server and client.

Other related works like [76, 77] proposed a framework for the QoS provisioning based on per-flow routing. The work in [76] presents a control logic that contains online and offline components. The online component sets up the route of the traffics for quality service, while the best-effort traffics are routed over multiple links. The offline component, on the other hand, runs continuously to optimize the load balancing across the network to meet the SLA policies. The framework solves the optimization problem that minimizes the maximum link utilization. The aim is to increase acceptance ratio for QoS requests. The experiment test was simulated using Python program, however it is unclear how the integration of OpenFlow-enabled network (such as the OpenFlow protocl and the flow-table statistical data) is taken place under this tool. Haitham Ghalwash et al. [77] proposed a framework for the QoS guarantee under SDN-based network. The framework is composed of monitoring and route determination modules. The algorithm tested for the route selection is either the shortest-path or QoS-enabled route selection. The experiment setup was based on mininet network emulator and OpenDaylight SDN controller. The fat-tree topology was employed to evaluate the performance of the proposed solution. The results show a reduction by the overall average delay, packet loss and jitter. Similar works in [74, 75, 78–82] adopted similar principle of per-flow routing. For example, the work in [78] uses the SDN-based architecture to improve the performance of IP video surveillance networks. By this, the framework performs bit-rate adjustment and rerouting in order to allocate more resource for quality services. The framework is tested using mininet network emulator and Ryu controller. While [79] proposes a framework of the QoS guarantee for streaming media. The framework employes the network calculus to model the QoS parameters of deterministic buffer size, effective bandwidth and delay. The performance of the framework was tested under mininet emulator network and Ryu controller. The topology for the experimental setup was composed of five switches.

On the other hand, other works in this area have employed the queuing and scheduling techniques to prioritize the flows in the SDN switch. The principle is similar to the DiffServ technique, in which flows are classified according to the application priority and prerequisite, then the flows are forwarded accordingly to the intended queues. Though the approach presents inevitably the problem of performance loss due to the class discretization and the fixed number of preconfigured traffic classes [83]. The work in [84] presents a system framework for QoS guaranteed technique under SDN-based network. The work implements a queue scheduling technique on SDN switch. It handles the application flows into different queues with different priorities. The study was evaluated under the theoretical and experimental analysis, though the experiment topology was composed of only one SDN switch and two hosts. Similarly, the work in [85] proposed a framework with classifiers and queuing schemes. The flow classification is used to associate the flow to the proper service class. While the queuing mechanism is used to allocate resources for certain traffic types. In order to realize it, multiple queues are set and assigned to different priorities. Each queue is associated to certain service class. The performance of the method was evaluated using OpenvSwitch SDN switch and Floodlight controller. The topology is SDN-based network of three switches that are connected to each other.

The work in [86] proposes a framework based on queuing technique for QoS provisioning over SDN-based networks. The framework reroutes the high priority flow by route optimization algorithm when the congestion is detected, however if no feasible route is available then the queue is enabled for the flow. The scheme sets and assigns queues to traffic depending on the priority level. The framework was evaluated using mininet emulator network and Ryu controller. The authors in [87, 88] propose a video over SDN architecture to support end-to-end QoS for video applications. In the proposed solution, the SDN controller pulls the data plane periodically to build its resource and topology databases. With the help of the centralized database, the controller selects an optimum path between the two ends. Alongside, they introduced solely a traffic management architecture for the SDN switch to prioritize the QoS packets properly along the optimum path. The test was implemented using ns3 tool. Other work in [89, 90] presented model-based QoS provisioning framework under SDN-based network. The work uses network calculus theory to define network model. The network models are defined: multi-hop model and threshold-based model.

The work in [91] proposes a meter-based QoS algorithm for multicast. The study uses a learning-based method to adopt intelligently the limit of packet rate. When link congestion occurs, the low priority traffics are routed among a number of different paths to release the congestion on the common link with multimedia traffic. The performance of the proposal was tested using a realistic SDN environment and Ryu as an SDN controller. The topology is composed of three OpenFlow switches, two servers and three hosts. Similarly, the authors in [92] propose a traffic shaping management for the SDN network. The application reduces the bandwidth of a given link in order to offload certain data traffic.

Among other solutions to enable QoS provisioning is the resource reservation for QoS-based applications. The work in [93] introduces an architecture defines certain number of different queues with a rate shaping in the SDN switch. The flows are classified based on the application priority and they are forwarded to the appropriate rate shaping queue to satisfy the business needs. By this, the resource reservation is realized through shaped virtual links. The implementation was based on OpenvSwicth OpenFlow-enabled switch and the POX SDN controller. The topology is composed of two SDN switches to validate the concept. The study was aimed for home broadband access networks. While the work in [94] presents a QoS framework using SDN architecture. The framework configures three queues for different priorities, two queues for the high and medium priority applications while the third queue dedicated for the best-effort traffics. The framework reserves resources for high priory flow at the ingress switch. The work was evaluated using Ofelia testbed and CityFlow project to validate the concept.

The work in [95] presents a resource reservation scheme for the end-to-end QoS provisioning under SDN-based networks. The proposal presents a new signaling mechanism for the resource allocation between inter-domain SDN networks. When the admission control accepts the traffic request, the controller setup the path in the network. The aim of the work is to solve the interoperability problem in provisioning end-to-end guaranteed service. The implementation was taken place on ONOS controller, a specific type of software implementation of the SDN controller. Similarly, the work in [96, 97] integrates the Resource ReSerVation Protocol (RSVP) protocol in the SDN controller for messaging exchange between the RSVP provider and requester. It utilizes the protocol to adjust the network performance through resource reservation and traffic prioritization. The performance of the concept was evaluated using the mininet network emulator and POX SDN controller. The topology is composed of three SDN switches, one SDN controller, two hosts and two servers.

Other work like [98, 99] presents a network architecture to ensure the resource reservation for the intended flows with the help of token-based authorization. By this, the SDN controller is used to orchestrate and automate the resource reservation in the network. The framework is composed of orchestrator to handle the requests and managing the network resources, while another controller is responsible for the advance reservation and it connects the sites between each other. The experiment was made using the ESNet SDN testbed that contains two sites while each site has two SDN switches and Ryu SDN controller. Similar work is done by [100], the reservation is established through network snapshots that represents the time during which resource availability remains constant. When a traffic request arrives, a list of possible paths which includes available path is calculated. The solution has been validated using minimet network emulator and ONOS SDN controller. The ESNet topology was employed for the experimental setup.

As a summary, the previous research in the area of QoS provisioning is divided mainly into three groups as summarized in Table 3.1. The first group uses the optimum dynamic routing method to offer QoS provisioning over SDN-based networks. By this approach, the flows are dynamically rerouted over time in order to cope with the business needs. The approach can control specific flow needs in fine-grained control way. The advantage of this approach is the network resources are not assigned directly over unlimited time to the group of flows. Instead, the resources are utilized in a more efficient way to adopt the network state with the business needs. Despite the benefits, the approach comes with difficulties associated to the need of high intelligence and observation in order to route the flow cleverly. With respect to SDN network, the complexity of control is shifted to the high-level management in the application layer in order to route the flows dynamically and intelligently over the network.

The second group utilizes the queue and scheduling management with optimum routing to reach improved delivery of the high priority traffic over regular traffic. By this approach, certain group of flows can benefit from the queue configuration. However the disadvantages by this approach lie in the needs of realizing the packet scheduling mechanisms on every SDN switch in the network domain. With respect to the SDN network, the OpenFlow is still in the early phases of development. Furthermore, another disadvantage is introduced by the classification and prioritization according to group of certain flows. This presents inevitably the problem of performance loss due to the class discretization and the fixed number of preconfigured traffic classes. Finally, the third group investigates the potential of employing resource reservation principle in QoS provisioning for SDN networks. By this, the SDN controller creates virtual network slices. The approach provides the capability of certain group of flows to meet the business needs, however the approach becomes unpractical to set on real network due to the limited network resources and high number of traffic flows.

Table 3.1: Table of comparison with different approaches for the QoS provisioning in SDN

Networking Framework	**Objective**	**Proposed Solution**
Flow Routing [68–82]	Optimum path (e.g. low delay and packet loss) are selected for the flow of quality services.	QoS flows are rerouted dynamically to satisfy the requirements of quality services.
Queue and Scheduling Management [84–92]	Queues are allocated for the class of service.	Flow classifiers identify the flow for certain class of service. According to the type of flow, flow packets are forwarded to the associated queues depends on the priority.
Resource Reservation [93–96, 98, 100–102]	Dedicated network resources are selected for QoS-based flows.	Resources like link bandwidth are explicitly reserved for certain flows.

3.3 Policy-based Network Management for SDN

With the evolution of SDN, many researchers have addressed the various challenges of the SDN concept and studied further its benefits. Unlike the traditional network, the SDN network emerged with unique features such as centralized approach, virtual network, interoperability and predefined interface between the control and data layers [8]. In the literature, number of works have studied and discussed the use of policy-based network management in the SDN architecture. By definition, the policy-based is a technique used to simplify the process definition of applying the constraint rules on the network underlying layer. Various applications can be realized with the use of policy-based management depending on the research sectors such as QoS provisioning and security. The work in [103] proposed a general combined architecture based on the SDN and PBNM concepts. It discusses the coexistence between the two networking technologies and it shows how SDN and PBNM can benefit from each other. However the study discusses the coexistence from the theoretical point of view and there is no experimental setup are performed to evaluate the architecture in a practical way.

A group of researcher focused on the integration of policy-based management under SDN for the autonomic QoS provisioning. For example, Bari et al. in [104] have addressed the use of PBNM under SDN. The authors present an automatic QoS policy enforcement framework for SDN called PolicyCop. The framework monitors the network parameters and adaptively reacts upon the detection of a policy violation. The flow of decision making considers the routing management as a main element for the policy enforcement. The test was evaluated using five OpenvSwitch SDN switches, floodlight SDN controller and four hosts that form a small size network topology. The experimental results consider a primitive use-case in terms of the traffic load and topology size for the throughput and link failure scenario, where the rerouting of QoS path is only considered at a solution. Further details on the monitoring scheme that PolicyCop uses can be found in [105]. Benet et al. introduced the work in [106] an SDN-based framework for the policy-driven data center interconnection. The work applies the SDN to enhance policy-based routing for Ethernet VPN-based data centers. In order to meet the SLA, the framework translates the high-level policies to low-level network configuration, the bandwidth reservation is taken place by the queue configuration, however the queue is set the bandwidth allocation for the entire aggregated traffic, instead of making per-flow reservation. The framework is evaluated using the fat-tree topology to demonstrate the geo-distributed networks. The framework was implemented on the OpenDayLight SDN controller. The scope of the study was focused on the application of SDN and policy-based management for other scale of network that is the interconnection between the data centers by using the Ethernet VPN method. Other works in [107, 108] combine the autonomic network mechanisms with the QoS management for SDN. The authors presented an extension for OpenFlow and OF-Config protocols to fulfill the needs of integration. To summarize, the aforementioned research works share a common goal of defining an architecture for the dynamic QoS network management under SDN-based network.

The work in [109] presented an autonomic QoS management architecture based on SDN-based network. The architecture reroutes the flow through the less congested route in the network. The framework is equipped with threshold-based policy so that it enforces the low-level policies when the threshold is violated by the measured values. The performance was evaluated using Mininet network emulator, Floodlight controller and OpenvSwitch switch. The results show that the approach optimizes the use of resources and improves the performance of QoS. Other work like [110] describes the integration of monitoring, policy, SLA management in the SONATA NFV service platform. The work extends the management and orchestration MANO framework that is used for the 5G network development. It aims to improve the service platform for autonomic policy-based network for the SONATA platform. On the other hand, the policy driven framework has been applied widely in the network security applications [110–112]. For example the work in [112] proposed a policy conflict detection in order to ensure the policies are conflict free for the security applications. Avoiding policy conflict means a policy does not invalidate another policy. Similarly the work in [110] presented an integrated security architecture for the SDN-based enterprise networks. The proposed architecture deals with the security attacks and it adopts the network dynamic accordingly. While, the work in [113] presented a policy enforcement framework over SDN network and it guarantees the consistency between the defined policies and the network behaviour. The work in [114] proposed a mechanism for automatic enforcement of security policies under SDN network. By this, it defines a policy refinement for translating the high-level requirements to low-level settings in the network. The work does not consider inference reasoning to detect policy conflict. The work in [115] presents a policy refinement for security application. The framework allows network operators to describe security policies and the system implement and refines them across the network.

In general most of the proposed approaches in the area of autonomic QoS provisioning have employed only the rerouting method as a measure when a policy is violated. The studies have not considered the comparison between the rate limiting and rerouting measures against the default SDN-based network. By this, the impact of the feedback closed-loop between the underlying SDN switches layer and the application layer was not studied. On the other hand, the test scenarios were limited in terms of topology size and traffic load including traffic types. While the evaluation metrics were limited to the QoS parameters and it is not extended to the expectation of the user. Here one contribution of this research aims to assess the routing and rate limiting methods against the default SDN-based network. The study uses metrics such as the PSNR and MOS values in order to evaluate the expectation of video streaming by the user. In addition to that, the experiment was evaluating towards a network mimic a realistic environment. In this research, the contribution in 1.4 proposes a complete QoS provisioning framework for the policy-based network management over SDN-based network. The contribution is the first step towards an intelligent SDN framework for routing control using data driven reinforcement-learning. Moreover, the contribution is a step towards applying a realistic test environment using video streaming and HTTP/FTP traffic mix.

Other related works introduce a programming language for the policy management. The policy language is used to define high level policy rules in the network. By definition, the policy rules is composed of a set of conditions and a set of actions to perform if the conditions are met. Kim et al. [5, 116] propose an event-driven network control framework using a high-level policy programming language. Similar work is presented in [117] by introducing a policy language. Han et al. [118] presents a layered policy management for SDN. However, the study mainly focuses on the detection mechanism and resolution strategy of different policies between various application domains such as load balancing and firewall. Similarly, other work in [119] integrates the PBNM into SDN and NFV networks. In the proposed approach, the PBNM solution is employed to design and manage service chaining in NFV-based network. It enables the network administrators to write SLAs using the policy language in order to guide the building of service chaining graphs. The work introduces a controlled natural language to create requirements and contrivances for the policies writing. The work in [120] developed a language for policy driven security architecture. The security policy language controls the flow of information according to defined constraints between end users across multiple SDN networks. For example, the unauthorized flows across multiple domains such flooding and spoofing are detected by the security policies defined in different SDN Controllers. Similarly, other work like [121] proposed a formal verification of security policies.

Other group of study investigated the policy refinement. The refinement process involves stages of decomposition, operationalization, deployment and re-refinement, and operates on policies expressed in a logical language flexible enough to be translated into many different enforceable configurations [122]. Machado et al. [123, 124] introduces a policy refinement framework over SDN. The policy authoring is based on logical reasoning. Whereas the QoS management is based on the routing decision only. Although the framework is used for high-level policy refinement based on logical reasoning, the solution did not investigate techniques for detection and resolution of policy conflicts. The policy refinement is used to translate the policies at the SLA level to the SLO level. However the configuration of SLOs on the lower-level network can be realized by different approaches like rerouting or queuing. The scope of this research work is to perform a study on the intelligent implementation of the SLO agreement on the network low-level. Similar work was done by [125], they introduce a policy authoring framework with logical reasoning for the SLA goals and to automate the refinement and the work in [126] introduces a policy refinement in cloud infrastructure.

As a summary, table 3.2 illustrates the different approaches of PBNM under SDN-based network. The first group presents the autonomic policy enforcement. The advantage by this approach lie in handling the network misbehavior automatically without the interaction from the network operator. Despite the benefit, the framework design becomes challenging and complex to address the policy validation and enforcement in an automatic way. On the other hand, the second group introduces languages in the network policy. The approach draws benefits to the network operator by configuring the network elements based on a high-level policy language. By this, the network operators do not interact with the lower details of the network devices. However the difficulties arise in defining a network policy language that represents flexibly a wide range of policies and at the same time formal enough to support automatic translation to logic [127]. While the third group presents the policy refinement. The benefits are observed in the automatic translation of high level requirements into many different enforceable configurations. However the problem of policy refinement is that of automating the movement from a high-level, abstract characterization of a policy, to policies in the languages of the various enforcement points [122].

Table 3.2: Table of comparison with different approaches of PBNM under SDN-basd network

Networking Framework	**Objective**	**Proposed Solution**
Autonomic Policy Enforcement [109–113] [104, 106–108, 112]	Configuring the network adaptively without the intervention of administrators, when the change of network state does not satisfy the business needs.	Introducing network monitoring with the classical PBNM system shall realize the autonomic network. The enforcement of high-level policies into the lower-level network configuration is realized through measures like traffic flows rerouting.
Network Policy Language [5, 116–119, 119–121]	The need for a unified network configuration language.	The definition of arithmetic function, constraints and events are the basis for constructing the policy language.
Policy Refinement [114, 115, 119, 123–126]	Automatic translation of high level requirements into enforceable configurations.	Techniques like the logical reasoning are be utilized for policy refinement.

3.4 Artificial Intelligence in SDN

With the evolution of SDN architecture, the southbound and northbound interfaces are introduced in order to program the underlying network layer. Based on such interfaces, the intelligent-based network solutions can be centralized in software-based SDN controller and the decision that is met by the AI solution can be configured easily on the lower-level network switches. One of the biggest challenges is the integration of intelligence solutions on the traditional networks due to the distributed view on network nodes. By SDN architecture, the feature of a centralized global view brings benefits to the AI applications for the decision making. Several studies in [128–133] present surveys on the application of AI techniques over the SDN paradigm. The research studies used the AI techniques to solve several problems related to the load balancing, security, and traffic engineering. They 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. In general, the ML techniques bring wide benefits such as to learn and understand the hidden knowledge behind the large data set that generated by the dynamic environment. However, the integration of ML technique comes with challenges like the design complexity [134].

ML is seen as a promising solution for the next generation networks. In the meanwhile, ML has been applied widely in the networking applications, such as traffic classification, network routing and security monitoring. The work in [135] surveyed the ML and deep learning methods used for traffic classification and prediction under SDN-based network. In practice, most of the applications classify the traffics according to the port numbers and addresses. However the work in [136] applied semi-supervised ML to classify the traffic flows into different QoS-aware classes. The deep packet inspection technique is used to detect the traffic based on the service type. By this, the traffic flows can be treated adequately to allocate more resources for the QoS flows. The solution was evaluated using a real internet data set. The success of the solution is compared with other algorithms such as K-means algorithms. The proposed solution outperforms the K-means algorithm by reaching a testing accuracy of 90%.

The work in [137] presents a traffic classification architecture based on SDN. The study uses the supervised learning to classify traffics into various applications like Youtube and Facebook. The results shows a high accuracy classification using supervised learning. It reaches a classification accuracy of 90% for the Facebook data, while 73% for the Linkedin data traffic. While the work in [138] presented a neural network estimator for application recognitions of flow resources. The algorithm finds a model to match the training traffic patterns. The aforementioned studies introduced traffic classification methods with high accuracy results. The solution was evaluated using one software Open-flow enabled switch, one controller and two hosts. Another work by Mestres et. al. [139] introduces the concept of knowledge-defined networking. The study discusses the integration between SDN, network analytic and ML to ultimately provide automated network control. Additionally, the study presents a set of specific use-cases that utilizes the benefits of knowledge-defined networking such as routing and resource management.

Furthermore, ML technique has been applied for the intrusion detection system under SDN-based networks [140–143]. The goal of the works is to recognize the network attack patterns by using various

ML techniques. For example, the work in [144] proposed a deep-reinforcement-learning-based QoS-aware routing protocol that focuses on the security threads. Two kinds of attacks on the switches are considered in the study, they are gray hole and Denial-of-Service attacks. The system decides which path is available to be assigned with the goal of avoiding malicious nodes.

Other kind of research is investigated by several researches to apply the RL for network routing. The direction of research for the application of RL in network routing is in line with the objectives and scope of this research. Therefore this section focuses on this kind of research topic. The research in [145] proposed an adaptive video streaming method based on Q-learning approach. The method determines the right time to re-route the traffic in SDN-based network to minimize the packet loss, quality changes and controller cost. The experimental results showed that the proposed system outperforms other algorithms such as the shortest path routing and greedy-based approaches. However, complex scenarios such as the large-scale topologies and high traffic volume were not addressed in the study. Similarly, Sendra et al. [146] proposed a distributed intelligent routing protocol for SDN-based network. The proposal uses reinforcement learning algorithm to find the best route that returns the most reward value. Through the learning process, the algorithms assigns different weights periodically to the paths and by learning it finds the best weights that leads to better reward. However, the calculation of cost function was not specifically defined in the study. The test was carried out using the mininet network emulator.

Whereas, Lin et al. [147] proposed a Q-learning based routing algorithm in a multi-layer hierarchical SDN environment. The SDN-based architecture is composed of recursive hierarchical control plane with three levels of controllers. Each domain controller is responsible for the routing inside its own domain. The domain controller determines the path inside the own domain, while the super controller determines the global path among the domains in the entire network. For the reinforcement-learning, the rewards function is based on the delay, packet loss adn available bandwidth. The reward function is described by a reward value closes to one means the link selection is preferred, while the value closes to negative one as the penalty. The proposed framework is evaluated using Sprint GIP network topology with multi-layer hierarchical architecture. The work in [148] proposed a RL-driven for routing management under SDN-based network. The study develops a system that addresses situation-aware and intelligent network routing management. The framework is composed of two modules, the first module responsible of continuous monitoring the network state based on the QoS metrics such as packet loss and delay. While the second module performs intelligent routing optimization task by using the reinforcement learning. The QoS-aware scheme is developed to detect and prevent network problems such as congestion and link over-utilization. The work was evaluated using mininet network emulator. It uses POX as a type of SDN controller with OpenFlow-enabled switch. To evaluate the performance, two normal scenarios without congestion is performed and a third scenario with congestion scenario. The proposed method is compared with Dijkstra algorithm-based method and the results showed that the proposed approach outperforms the Dijkstra algorithm-based method.

The work in [149] explores ML for selecting the least congested path for routing in SDN-based network. Two methods were proposed for path selection from a list of possible paths to route the traffic flow, K-means clustering and vector space model with cosine similarity. The network state is represented by weights on the links, in which at the end of training phase a number of cluster set are generated. The set indicates the best paths in the network to be candidate for selection. The evaluation was made against the conventional network based on Dijkstra's algorithm where cost for each link set to one. The results showed that cosine similarity outperforms other variants. The work is evaluated using mininet network emulator and Ryu SDN controller, while the topology is composed of 5 SDN switches. On the other hand, the work in [150] proposed a deep reinforcement learning for routing optimization under SDN-based network. The approach utilizes the off-policy and actor-critic deep learning method. By this, the state is represented by the traffic matrix based on the bandwidth request and the action is the path, while the reward is the average network delay. Though the approach tends to minimize the network's delay, however the state and action space can be exploded with the network size and number of flows. The approach was evaluated using the OMNeT++ simulator with a network topology of 14 nodes. The study discusses as well the advantages of reinforcement learning compared to approaches like approach with the heuristics methods.

As a summary, table 3.3 illustrates the type of network applications categorized according to the study objective and solution. In the table, the first group presents the introduction of ML techniques for the network traffic classification. Based on the classification of traffic, the network providers are able to detect the malicious attacks, reallocate network resources, and perform traffic modeling [151]. Traditional traffic classification was port-based classification, however nowadays the growing of traffic volume with data encryption raises the challenges for designing accurate traffic classification. On the other hand, the second group presents a special case of traffic classification of the first group. The research investigates mainly the techniques such as the signature-based and anomaly-based detection to identify the malicious activities in the network. Finally, the third group investigates the ML techniques for the network traffic routing. The work of this research is in line with the third group. The research objective aims to integrate the AI solution for network routing in SDN-based network. In most of the current research, the objective is to invent a new routing algorithm with the help of AI techniques. However the research here utilizes the RL-based method to use the existing routing algorithms and find the best selection of routing algorithms to fit the network needs. Moreover, most of the current researches uses test setup based on limited scenarios such as single network topology. Though the research here evaluate the proposed solution on different scenarios in terms of traffic load and network topology.

Table 3.3: Table of comparison with the different approaches of AI/ML techniques for SDN

Networking Application	**Objective**	**Proposed Solution**
Traffic Classification [135–138]	Traffic are classified in general according to the applications.	ML learning methods (e.g. Nueral Network estimator) are utilized to train the model for the type of application traffic.
Intrusion Detection System [140–144]	Traffic belong malicious activities to be identified.	Predicting network attack patterns by using ML learning algorithms (e.g. deep learning).
Network Routing [144–150]	Enhance the performance of quality applications (e.g. multimedia traffic).	Reinforcement learning, as an ML algorithm, is used to drive intelligently the traffic routing in the SDN network.

3.5 Remaining Challenges and Open Issues

Recently researchers have shown the wide studies of QoS provisioning and policy management under SDN-based network. However, there are still a number of research challenges and open issues that need to be further studied. The QoS policy management under SDN-based network remains a challenging topic. This section outlines some of the remaining challenges related to this research:

- **Including Artificial Intelligence Techniques for SDN Network Management:** The lack of intelligence meaning in the decision making of policy management represents one of the fundamental issue. The current researches use the predefined pick of actions for a certain condition. For example, when a policy violation is detected, measures like finding the optimum path is triggered primitively to reroute the flows on alternative paths. The intelligent aspect based on learning or modeling is not involved to choose the adequate measure for resolving the network conflict. An approach to tackle this problem is the use of Reinforcement learning to enforce the learning of past experiences and modeling to gain a long term reward. The aim of this research is to introduce the intelligence inside the PDP component of PBNM framework, while utilizing the key features of SDN architecture. Alongside, the design by itself of the RL algorithm is certainly a fundamental challenge.

- **The Utilization of SDN Key Features:** The utilization of SDN features, for example the centralization, the visibility and the programmability, is considered one of the most important issues. A research challenge comes from the design of the network functional component under the SDN infrastructure. For example the dispatch of resilient and consistent data to the upper layer of making decision. The study presents at how the integration solutions between the four aspects QoS, SDN, PBNM and ML are utilized in a single network management framework.

- **Self-organized Network Management:** Another challenge to be addressed is the design of the closed-loop control structure between networking monitoring, decision making, and agreement violation in order to draw the maximum benefits for the perspective of autonomic network management. A self-organization concept shall be realized in the underlining network devices. In general, the network shall adjust its behavior according to the network state. If a state breaks the business agreement, the network control management shall make the change to have a state that complies to the needs.

3.6 Chapter Summary

This chapter aims to present the related works done in the area of SDN-based networks. It introduces the current industrial solutions for driving SDN solutions into the networking commercial market. Various research groups have investigated the QoS provisioning under SDN-based network. Routing, queuing management, and explicit resource reservation were among the current research topics for the QoS provisioning. This chapter also described the existing research on policy management solutions. For the autonomic policy scheme network, network monitoring becomes an essential network component. The QoS policy management provides benefits for the QoS management by operating networking applications under the SLA agreement. The low-level policies are created to adhere the rules in high level contract between service providers and customers. On the other hand, AI and ML added motivation for the research community to integrate intelligent applications over SDN. ML techniques has been applied widely for the applications like traffic classification, network security and routing optimization in the academia field. Finally, the chapter ends up with a summarize of the current remaining challenges and opening issues related the QoS policy management under SDN-based network.

This research presents entirely a complete framework for end-to-end QoS provisioning under SDN-based network. In order to achieve this, the research combines three main elements into single framework. These elements are: SDN, AI, PBNM, in which every element is taken advantages of the other. For example through the centralization and programmability of SDN network, it becomes viable to implement a complex AI solution in a centralized way. On the other hand, the AI techniques brings various benefits of accommodating algorithms to cope with a complex networking problem. PBNM provides a management solution to manage network services that meet compliance needs. Though via the centralization, controlling and monitoring in SDN, a self-organized network for end-to-end QoS provisioning becomes visible. The aim of this research is the notion of bringing SDN and PBNM together with AI techniques to networking.

The novelty of the proposed framework is reflected in several directions along the research. A novel scheme is introduced for accommodating the compression techniques at the data collection level of monitoring traffic data. The sparsity approximation algorithms are employed to compress the aggregated data in the SDN switch, while the SDN controller recovers the sparse data. According to several existing studies, the impact of this approach is to reduce the bandwidth usage by network monitoring along the link between control and forwarding planes. The proposed method offers an increase of the network observability by reconstructing the compressed data to a long time series of original data. This method helps to reduce the amount of exchanged data in terms of amounts of bytes between the controller and switches if the SDN controller is embedded inside the network while sharing the same network resource of link bandwidth with the switches. Such an approach reduces the network link utilization by reducing the overhead of monitoring.

Section 4.2.1.3 illustrates in detail the metric performance used for evaluating the method. Moreover, in this research an innovative probabilistic routing is proposed. The solution utilizes Bayes' theorem in order to determine the link probability. This in turn is used to select the route that satisfies the given bandwidth constraint. The impact of this contribution is to enable the routing algorithm to calculate the routing path when less information is advertised by the switch plane. The impact is to reduce the overhead on the link between control and forwarding plane in an SDN-based environment. The method uses the link probability to compute a path that satisfies the given bandwidth constraints as described in section 4.2.2.

Furthermore, a novel approach is presented in section 4.4 by integrating RL in the routing decision making module. The RL-based approach selects the most appropriate routing algorithm from a set of algorithms while maintaining the flow satisfaction with respect to the defined SLA requirements. In contrast to the existing studies, the proposed approach enhances the other state-of-the-art routing algorithms in terms of performance metrics such as: higher average throughput, lower average packet loss, better average PSNR and MOS. Moreover, the proposed method offers a test bench to evaluate the method against the existing solution using an experimental setup under a realistic SDN environment and the work studies the impact of network topology and traffic load.

Chapter 4

Proposed System Architecture and Algorithms

This chapter introduces the proposed conceptual framework and operational definitions. It starts with the description of the overall proposed system architecture, followed by the details of the three main contributions of this research, such as: (1) measurement collection and probabilistic-based routing, (2) policy-based QoS management framework, and (3) RL-based Decision Making for Routing Algorithms under Policy-based SDN Environment

4.1 Overall Proposed System Architecture

Figure 4.1 shows the high level overview of the functional block diagram for the proposed framework. It illustrates the integration of the three element into one single framework for the purpose of end-to-end QoS provisioning. The framework is based on SDN architecture (as described in 2.2). Moreover it utilizes the three layers of PBNM architecture for (as described in 2.3) and it contains an end-to-end management module to determine the way to set up routes on the network while maintaining the SLA requirements. The end-to-end management module is realized by a decision making process based on AI principle (as described in 2.5). In general the functional process in the proposed framework works as follows: in order to establish an end-to-end QoS communication between two end users in the SDN network, the source sends first a request to the SDN controller to set up an end-to-end QoS path. Based on the network services, the controller on behalf forwards the message to the upper layer, in which the intelligence of end-to-end management is located. The management is comprised on intelligent decision unit to decide on the right action to apply that complies to the SLA requirements stored in the policy repository.

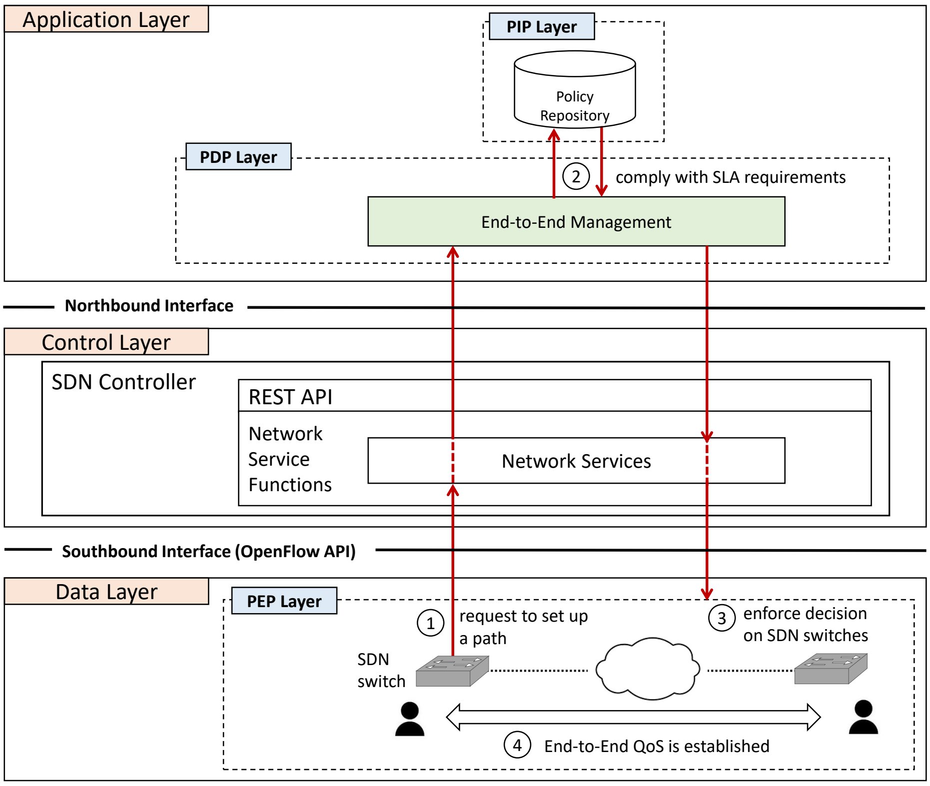

Figure 4.1: High level functional block diagram of the proposed framework

On the other hand, figure 4.2 illustrates the overall proposed system architecture as a high level view of the system. It shows the main functional block diagram of the proposed system, while the detailed description is presented in the upcoming sections. The proposed framework builds on top of the classical SDN architecture and adds three main contributions: (1) *measurement collection and probabilistic-based routing solution* (indicated by light green colored block), (2) *policy-based QoS management solution* (indicated by light blue colored block) and (3) *RL-based Decision Making for Routing Algorithms under Policy-based SDN Environment* (indicated by light orange colored block).

The measurement collection and probabilistic-based routing solution intends to reduce the monitoring overhead on the control link between the data and control planes while increasing the network state observability. The monitoring in SDN represents an essential engine, the application in upper layer needs to observe the underlying network condition and to check whether the network performance is complying to the business needs. On the other hand, the global knowledge of network state contains uncertainties, a probabilistic routing is proposed to overcome this problem by utilizing the probability theory in the routing computation. The proposed probabilistic-based routing is coupled with the monitoring component via link state update. In this thesis, the QoS provisioning over SDN is realized through a PBNM. The proposed PBNM-based solution, contains techniques for network monitoring and decision making. Finally, the integration of a reinforcement learning algorithm for routing management is proposed. It intends to maximize the network utilization and fulfill the QoS application requirements by finding the most suitable trade-off between throughput, packet loss and quality.

The three main contributions of the overall proposed framework are described in detail in the following sections.

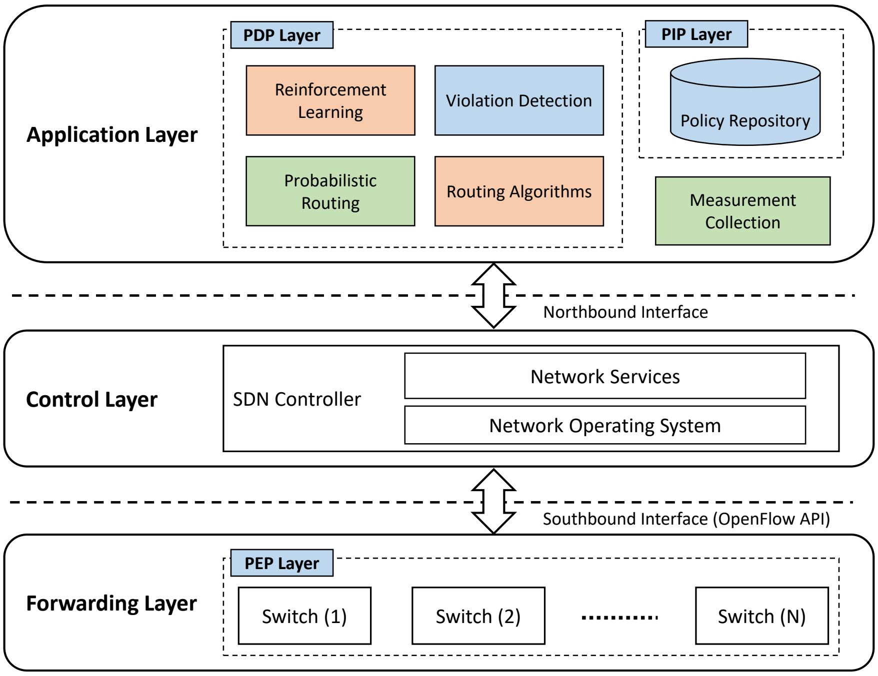

Figure 4.2: Overall proposed system architecture

4.2 Measurement Collection and Probabilistic-based Routing

4.2.1 Monitoring Techniques under SDN Infrastructure

Network monitoring represents one of the fundamental pillar in the autonomous and intelligent networks. In order to meet the policy agreement, network monitoring needs to observe the underlying network infrastructure and check whether its performance satisfies the business needs. In the meanwhile, the research area of SDN monitoring can be classified into two main categories based on how and where the measurement is taking place. The first approach utilizes the existing architectural design of SDN. While the second extends the internal architecture of SDN switch in order to achieve other goal of monitoring like high accuracy and observability, as well as low network overhead.

In general, there are two main mechanisms to observe the state of SDN: active or passive methods. They mainly differ in the way the network state is dispatched to the SDN controller. With the active method, the SDN controller injects query packets into the network in order to request the underlying network state. While no extra query packets are added in the passive method. On the other hand, the SDN switches push the status regularly to the controller. This section introduces the two techniques and it proposes additional concept for measurement collection.

4.2.1.1 Monitoring based on Continuous Measurement Update

OpenFlow mainly supports the request-reply technique for active monitoring. Under the continuous monitoring, the controller requests explicitly the flow statistics from the switches. As a result, the controller can build a centralized global view of the network state. Due to the principle of request-reply, the continuous update can add high overhead on the control link between the two planes if the statistics are frequently requested. In this thesis, the continuous measurement update was implemented in the proposed framework 4.3.

4.2.1.2 Monitoring based on Link State Update

Another approach to reduce the monitoring overhead is the use of link state update. Under this method, the switch publishes passively the statistics only when certain conditions are fulfilled. Generally, the network state changes over time according to several factors, such as: the failure of a network node or instantaneous traffic overload. In this case, only the most important performance profiles are published to the controller while other profiles are kept unpublished. The low observability of entire traffic profiles remain the main disadvantage of this method.

The *Link State Update Policy* consists of a threshold-based triggering policy combined with Hold-Down Timer (HDT) [152]. By this, HDT is used to push the statistics to the controller, if the threshold is not met during the time window of HDT. This ensures that the controller shall be updated with the flow statistics regularly. Thus, the link state update is triggered only if a certain threshold thr is exceeded or the HDT is expired. For the bandwidth availability, the threshold-based policy checks if $\left|\frac{b_{last}-b_k}{b_{last}}\right| > thr$ is met, where b_{last} is the last updated value of the available bandwidth metric and b_k is the current state of the bandwidth metric. To this extend, when the threshold and HDT are set to very low, the SDN switches push data to the controller frequently.

4.2.1.3 Measurement Collection Through Compression Technique

This research establishes a further study on monitoring scheme. It proposes a data collection scheme based on a compression technique for SDN-based network. Finding the solution for the trade-off between the network overhead and accurate network view poses a significant challenge. The motivation for utilizing the compression method is to reduce the size of data transfer over the control path, while it increases the network observability. It employs sparsity approximation algorithms for compressing the aggregated data in the SDN switch, while the recovery of the sparse data is taking place at the controller.

The proposed technique presents a data collection scheme based on the compression technique under SDN. Here the sparsity approximation algorithms are utilized to compress the aggregated data in the SDN switch, while the SDN controller recovers the sparse data. The approach aims at further decreasing the control link usage while increasing the network observability. In order to do this, it extends the existing functionality of the SDN switch by integrating the dictionary learning algorithms. The current method is applied for the throughput metric.

4.2.1.3.1 Sparsity Approximation Algorithms The research employs the K-SVD algorithm to train the dictionary, while the Orthogonal Matching Pursuit (OMP) algorithm is used for the sparse approximation.

4.2.1.3.2 The K-SVD Algorithm The K-SVD algorithm is an iterative algorithm used to train an over-complete dictionary D [153]. The training of a dictionary of finite K basis vectors is performed iteratively by solving the following minimization model:

$$\min_{D,\{x_i\}} \sum_{i=1}^{N} \|y_i - \hat{D}x_i\|_2 \quad s.t. \quad \forall i, \|x_i\|_0 \leq \epsilon \tag{4.1}$$

where $\hat{D} \in \Re^{M \times K}$ is the estimated trained dictionary and $K > M$ for an over-complete dictionary, $y \in \Re^M$ is the training data and $x \in \Re^K$ is the sparse approximation coefficients vector that contains non-zero components denoted by s. While ϵ is the maximum number of basic vectors in the sparse representation stage. The K-SVD algorithm iterates between two stages: (1) the calculation of sparse coefficients and (2) the dictionary update. Finding the dictionary solution that approximates the true minimum is NP-hard [154]. For K-SVD with OMP as pursuit algorithm, the computational complexity per training iteration is $O\left(sMK + MK^2 + M^2K\right)$ [154].

4.2.1.3.3 The OMP Algorithm

The OMP algorithm is an iterative greedy algorithm for sparse approximation [155]. The sparse approximation problem is modeled to find a linear combination of basis vectors from a trained dictionary. Having an over-complete dictionary $\hat{D}$ and an observation vector y, OMP algorithm finds the corresponding sparse approximation coefficients vector x with s non-zero components by fulfilling the following formula:

$$y_i = \hat{D} x_i \tag{4.2}$$

Finding the best approximation is an NP-hard problem, however OMP converges for the sub-optimal solution in no more than K iterations. In each iteration, the errors are minimized until it reaches the stopping criterion. The OMP algorithm has a computational complexity of $O(sMK)$ [156]. Table 4.1 shows the set of used variables in the problem definition.

Table 4.1: Notation Definition

Variable	Definition
$\hat{D}$	The estimated trained dictionary
x	The sparse approximation coefficients vector
y	The measurement or observation vector that is used for dictionary training
T	The set of training data for dictionary
K	The number of columns in the trained dictionary $\hat{D}$
M	The number of rows in the trained dictionary $\hat{D}$
s	The number of non-zero components in the sparse vector x
ϵ	The maximum number of basic vectors in the sparse representation stage
CR	The compression ratio

4.2.1.3.4 Proposed Monitoring Architecture

Figure 4.3 shows the high-level picture of component-based architectural design. At the forwarding layer, the SDN switch contains three main components: *Flow Aggregation*, *Dictionary Training* and *Sparse Approximation*. The *Flow Aggregation* is responsible to compute the overall link throughput metric. It aggregates the flow statistics that exist in the flow tables. In the *Dictionary Training*, the K-SVD algorithm estimates the dictionary $\hat{D}$ based on the observation of the training data set. To do this, the throughput data of one-dimensional structure is divided into sections of length M. When the dictionary $\hat{D}$ is predicted, the OMP algorithm in the *Sparse Approximation* is used to calculate the sparsity approximation on defined time window. By then, the data in the sparsity form is ready to be sent to the controller. To this end, the OpenFlow-enabled switch does not yet support this mechanism of data transfer and the proposed method is evaluated in the next chapter using Matlab. On the other side, the application layer contains two components: the topology component which maintains the current network topology. The recovery of sparse approximation is responsible to reconstruct the original data from the sparse approximation.

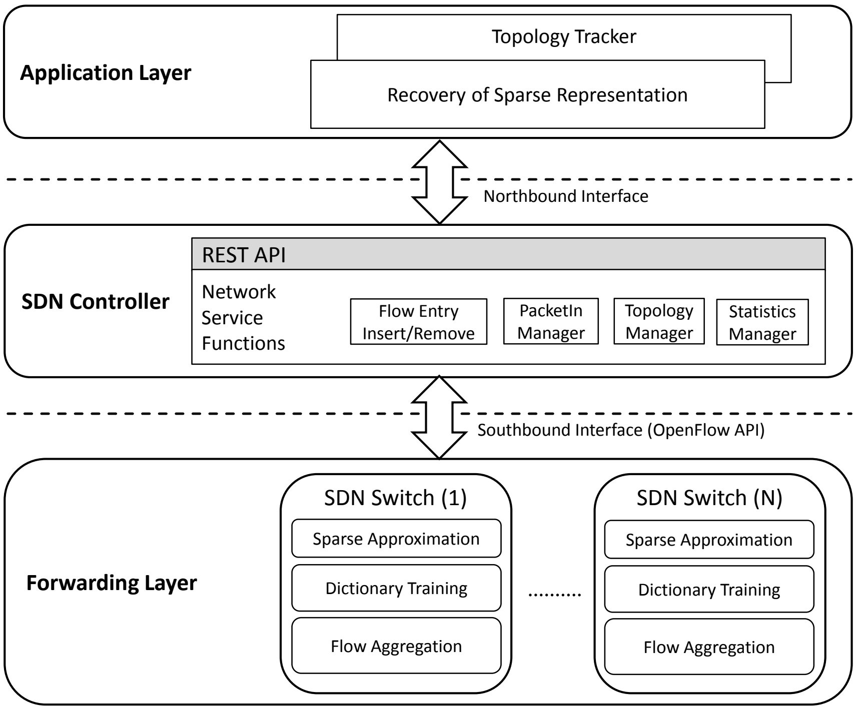

Figure 4.3: Foreseen Architecture of the Compression-based Monitoring

4.2.1.4 Methods Comparison

Table 4.2 summarizes the aforementioned approaches of network monitoring under SDN-based network.

4.2.2 Probabilistic-based QoS Routing Algorithm

The QoS routing approaches mainly rely on the instantaneous metric observations without the consideration of imprecision in the state information. However, in general, the global knowledge of the network state information contains uncertainties caused by various perturbation factors, such as the inaccurate metric measurement methods. Thus, the employment of probabilistic schemes into QoS routing shall be favorable to the traditional cost-optimization solutions. This section proposes BaProbSDN, a probabilistic QoS routing mechanism for SDN. The QoS routing algorithm employs the bandwidth availability metric as a QoS routing constraint for unicast data delivery. BaProbSDN

Table 4.2: Table of comparison with different approaches for network monitoring in SDN

Networking Monitoring	**Objective**	**Proposed Solution**
Continuous Update 4.2.1.1	Simple and adopted for variable monitoring interval. Frequent request for statistics can introduce high monitoring overhead.	Existing OpenFlow-enabled switches support the feature.
Link State Update 4.2.1.2	Only interesting measurement profile are published to the controller. It can maintain low monitoring overhead, however the network traffic observability remains low.	Later versions of OpenFlow support partial implementation of the feature.
Compression-based Update 4.2.1.3	High network observability with high accuracy and low monitoring overhead.	The OpenFlow protocol needs to be extended to support the functionality.

utilizes the Bayes' theorem and Bayesian network model in order to determine the link probability, in which it selects the route that satisfies the given bandwidth constraint.

Figure 4.4 illustrates the proposed functional units built on top of the classical SDN architecture. The figure shows the detailed functionality of a high description block (indicated by light blue colored block) in Figure 4.2. The approach exploits the benefits of SDN in terms of centralized management and configuration. The *Link State Update Policy* is integrated here so that the SDN switch pushes the link state information to the SDN controller every time a certain criterion is met. As each link has its own bandwidth capacity, the SDN switch computes the bandwidth availability metric through the *Metric Computation* unit. The application layer contains two blocks, namely the proposed *Probabilistic QoS Routing Algorithm (BaProbSDN)* and the *Global Information State Management* which maintains a consistent global network state information matrix.

4.2.2.1 BaProbSDN

BaProbSDN is a probabilistic-based QoS routing algorithm that utilizes the directed acyclic graph Bayesian network and Bayes' theorem to obtain the link probability of bandwidth availability. The probability is determined along a window of observation, referred to as Window Size (WS) (i.e., a window size gives the observation period that contains a number of measurement samples which are updated by the SDN switches). The following states are defined to model the bandwidth availability of a link as described by the Bayesian network listed in Fig. 4.5:

- $L \equiv b_k > b_{req}$: Does the link has enough bandwidth
- $\Delta B \equiv |b_{req} - b_{last}|$: The amplitude of last advertised bandwidth value b_{last} relative to the

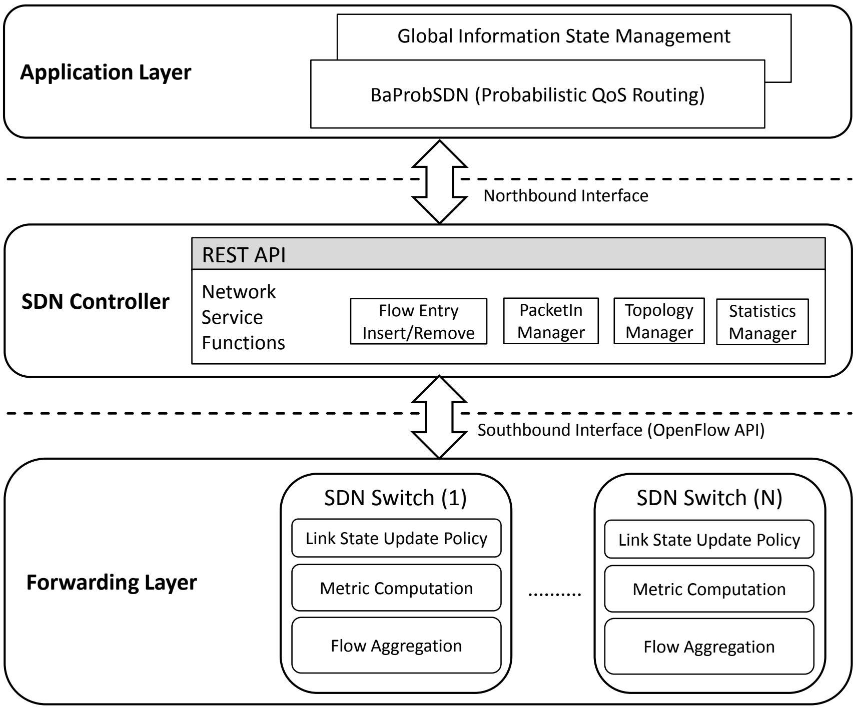

Figure 4.4: SDN architecture with proposed functional units

requested bandwidth b_{req}

- $T \equiv t\left(b_{last}\right) = i$: The temporal order of last advertised bandwidth $t\left(b_{last}\right)$ where i is the location of bandwidth measurement in the observation window

where b_k is the current state of the bandwidth availability and b_{req} is the requested bandwidth. While b_{last} is the last updated value of the available bandwidth metric. Given the above mentioned states, T and ΔB are independent but they become dependent when L is given, therefore the prior joint probability is defined as:

$$Pr\left[L, \Delta B, T\right] = Pr\left[L|\Delta B, T\right] Pr\left[\Delta B\right] Pr\left[T\right] \tag{4.3}$$

In order to find the probability of a link, it is essential to compute the posterior probability $Pr\left[L|\Delta B, T\right]$ which informs whether the link supports a bandwidth larger than the requested one when the last advertised measurement and the requested bandwidth are given. It is rather difficult to calculate directly the probability of hypothesis L given the evidence of ΔB and T, therefore Bayes rule simplifies the problem by considering the observables quantities as formulated in 4.4:

$$Pr\left[L|\Delta B, T\right] = \frac{Pr\left[\Delta B, T|L\right] Pr\left[L\right]}{Pr\left[\Delta B\right] Pr\left[T\right]} = \frac{Pr\left[L, \Delta B, T\right]}{Pr\left[\Delta B\right] Pr\left[T\right]} \tag{4.4}$$

The likelihood $Pr\left[\Delta B, T|L\right]$ is estimated from the system observations of a large training dataset in advance. Here the likelihood is calculated approximately by the means of a counting process. The proposed approach is based on Bayes' theorem and the causal relation in Bayesian network. Given the location of the last advertised bandwidth availability within the observation window and the value of current requested bandwidth, the model estimates the link probability based on this information. Basically, a link is more likely to support the request as the location of the last update approaches the request arrival time and the difference ΔB becomes larger. However, the model reaches moderate probability as the difference ΔB becomes smaller, this indicates that there is no evidence whether the link supports lower or higher bandwidth at the time of the arrival request. Table 4.3 shows the set of used variables in the problem definition.

The probability-based routing algorithm is formulated as a path finder of p^* that is most likely to satisfy the bandwidth constraint B, and the following condition is met for every other path p [157]:

$$\pi_B\left(p^*\right) \geq \pi_B\left(p\right) \tag{4.5}$$

where $\pi_B\left(p\right) = \prod_{(i,j)\in p} Pr\left[b\left(i,j\right)\right]$ and $b\left(i,j\right)$ is the residual bandwidth of the link $\left(i,j\right)$. The solution of eq. 4.5 is carried out by assigning to each link a weight of $-log\left(Pr\left[b\left(i,j\right)\right]\right)$, so that the problem is transformed to additive operation in path finding. To find the path $\pi_B\left(p^*\right)$ the standard Dijkstra shortest path algorithm is executed on the associated weights with a computational complexity of $O(L\ \mathrm{log}N)$. This algorithm is known as the Most Probable Bandwidth Constrained Path (MP-BCP) problem [157]. According to the observations, the model produces different probabilities depending on the order of the observation sequence. When a request arrives, the probability is determined by equation 4.4 for each link in the network. Therefore the group of links that shape a QoS path between the source and destination hosts can be calculated by solving the MP-BCP problem. The problem solution was modified so that the group of paths that most likely satisfy the QoS requirements are selected first. Then the path which contains the links with higher probabilities is chosen. Additionally for lonely threshold-based policy, the BaProbSDN algorithm can determine if the bandwidth availability that is not advertised by the switch supports enough bandwidth for the QoS request. This can be achieved by checking if the requested bandwidth lies in the range of $b_{req} \notin \left[b_{k-1}\left(1+thr\right), b_{k-1}\left(1-thr\right)\right]$.

4.3 Policy-based QoS Management Framework

In order to overcome the challenges of network traffic growth, the policy-based management proposes a solution to automate the process of network configuration via a set of constraint rules. The use of

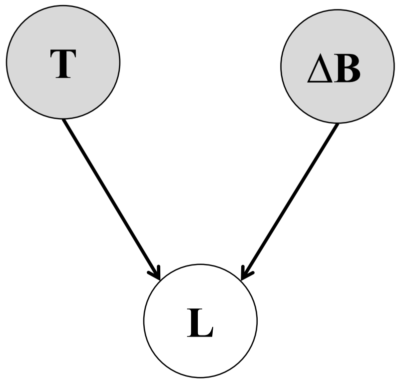

Figure 4.5: Bayesian network model in BaProbSDN for link probability

Table 4.3: Notation Definition

Variable	Definition
b_{req}	The requested bandwidth by the traffic flow
b_k	The current state of the bandwidth availability at time k
b_{last}	The last advertised value of the bandwidth availability
$b(i, j)$	The residual or available bandwidth of the link (i,j)
WS	The size of observation window
L	Indicator if a link has a higher bandwidth availability than the requested
ΔB	The amplitude of last advertised bandwidth value b_{last} relative to the requested bandwidth b_{req}
$T, t(b_{last})$	The temporal order of the last advertised bandwidth b_{last}
B	The bandwidth constraint that is met by the path
p^*	The path that it is most likely satisfies the bandwidth constraint B
thr	The threshold of the link state update policy

SDN and PBNM brings several benefits compared to legacy networks. On the one hand, the SDN concept reduces the management complexity through the centralization of the entire management. While the PBNM enables a simplified management of the data plane as compared to the complex middleboxes in the traditional networks [6]. It controls the resource provisioning in order to meet the business requirements.

This section illustrates the integration of policy network management architecture under the SDN-based network. It proposes a policy-based management framework over SDN for QoS provisioning. By this, the proposed approach monitors the QoS parameters of the active flows and it enforces dynamically new decisions on the underlying SDN switches in order to adapt the network state to the current demanded high-level policies. In the proposed framework, the continuous monitoring update 4.2.1.1 is realized. By using the loop chain approach between the network monitoring and policy validation/enforcement, the framework can achieve end-to-end QoS. Upon detection of a policy violation two flow management techniques are implemented, such as: *rerouting* and *rate limiting*.

4.3.1 Framework Architecture

The proposed PBNM-based SDN framework is illustrated in Figure 4.6 and consists of the following main components: **(1) Policy Repository** - stores the entire high-level policy rules that reflect the requirements for the agreed services between the service provider and customers. **(2) Topology Tracker** - maps the physical network elements into a graphical structure. The output of this component is fed into the QoS metrics monitoring unit to build a global image of the instantaneous network state. **(3) Admission Control** - accepts or declines network connections based on the availability of network resources. **(4) QoS Metrics Monitor** - measures the QoS metrics (e.g., throughput, packet loss rate and delay) of each flow in the network. The metrics are determined by periodically sending flow statistics query messages to the switches. The constructed view on the network load is used later by the violation detector to indicate the misbehaving traffic flows. **(5) Violation Detector** - represents the validation engine to release the necessary measures to converge the network to the state agreed by SLO requirements. **(6) Active Flows Tracker** - it tracks the active flow routes in SDN. The built routing table is utilized by the monitor unit to estimate the throughput per active flow. **(7) Route Manager** - computes the least loaded path demanded by the application's QoS based on the instantaneous network state. While it seeks the shortest path for the best-effort traffic. **(8) Rate Limiting Manager** - configures the rate limit parameter along the best-effort route.

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.

4.3.2 Network Management Function

The functional components of the proposed architecture design are mapped to the general three-level PBNM framework (PIP layer, PDP layer, and PEP layer) [41]. Two cases are identified for managing the network state: (1) *upon receiving a new route request* - Initially, the controller receives a packet-in message and the admission control decides whether to reject or accept the upcoming request based on the resources availability. If the request is accepted, the application type is identified, such that in case of best-effort request, the shortest path is determined using a method based on Dijkstra's algorithm. Whereas in case of QoS application, the least congested path is chosen. (2) *a policy violation is detected* - The case is illustrated in Figure 4.7. Initially, the network monitoring component collects

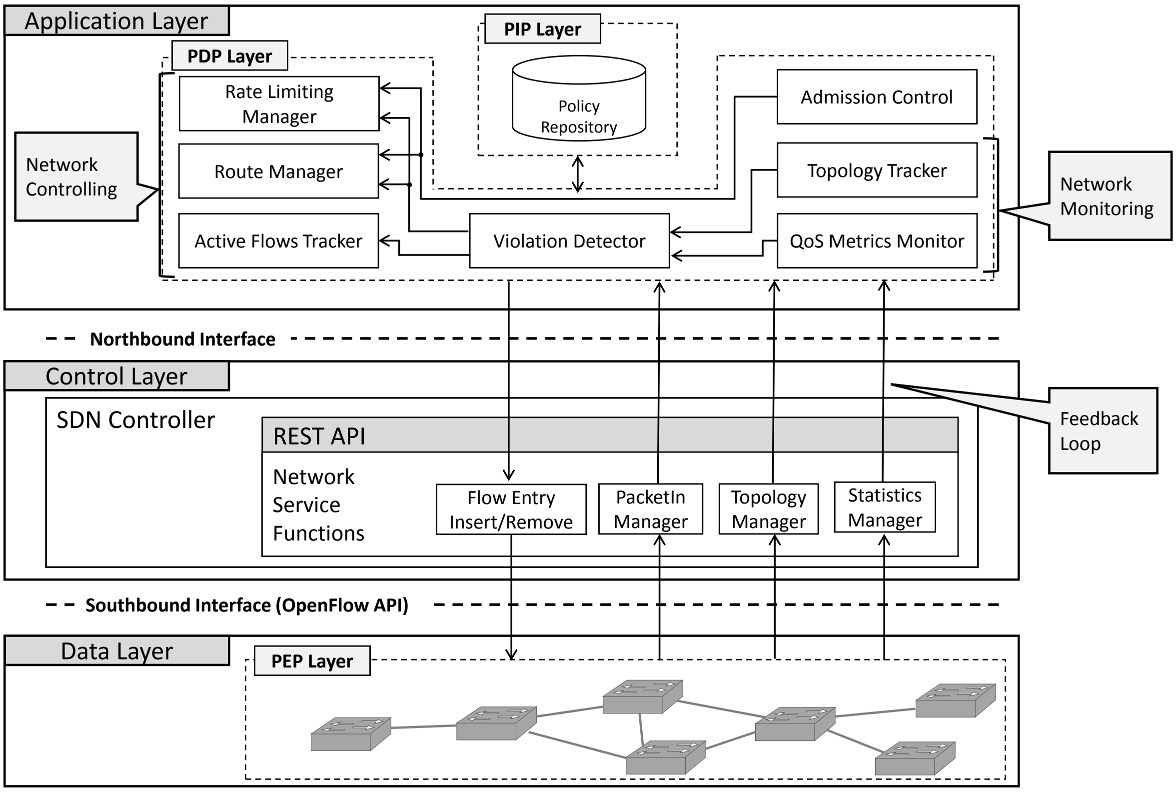

Figure 4.6: Proposed PBNM-based SDN Framework

periodically flow statistics from the switches. The measurements are used to calculate the available bandwidth and packet loss rate of all flows in the routing table. As a result, the controller builds a global view of the network load. The violation detector determines whether a high-level policy rule is broken. If the policy is broken, it identifies the flow that causes the violation by comparing the measured quality metrics against the high-level policy. This is further used to identify the congested link along the misbehaving (violated) flow that causes the violation. Upon a violation, the violation detector either trigger the route manager to choose an alternate route for the background traffic over other shortest path routes or it triggers the rate limiting manager to reduce the bandwidth budget for the background violating the best-effort flows.

4.4 RL-based Decision Making for Routing Algorithms under Policy-based SDN Environment

One of the most significant paradigm shifts within the networking industry is represented by the introduction of SDN. To this extend, SDN comes with key advantages in comparison with the traditional

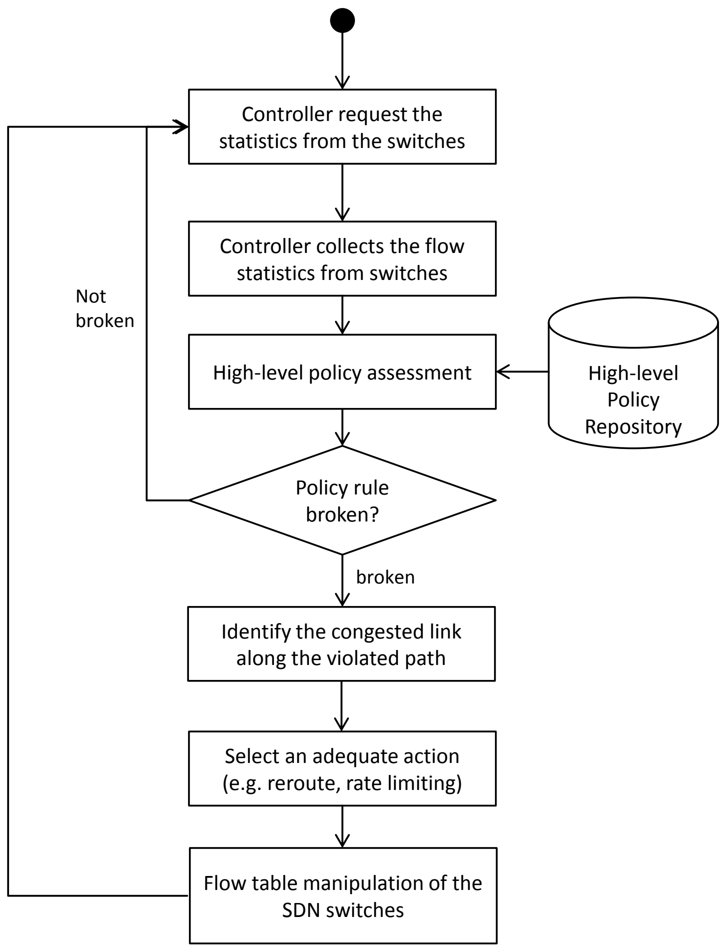

Figure 4.7: Network management work flow

network. Some of the features are centralization, programmability via standardized interface, the decoupling of control and forwarding planes, and feasibility through global network image [36]. These advantages encourage to centralize the RL-based solution on the application layer of SDN network. With SDN, network design becomes simplified as the entire network is controlled by a SDN controller using the open interfaces [37]. Additionally by introducing the closed-loop system through monitoring, controlling and validation, a self-organization concept shall be realized in general in the underlining network devices. The network shall adjust its state to comply with the business needs [158, 159].

On the other hand, artificial intelligence has recently gained increasing popularity due to its applications in almost every sector [160–163]. AI technologies bring many benefits when integrated into any system [48]. For example through the learning process, the system builds a knowledge to improve the decision making in a more efficient way. The objective of this research is to integrate the learning method in the decision-making unit of the policy management system. This work applies and utilizes the Q-learning method to decide on the most suitable conventional routing algorithm to be applied on the traffic flows. It is used to select the most appropriate routing algorithm from a set of centralized routing algorithms that maximizes the return reward from the network. Here for proof of concept, the set of routing algorithms is MHA, WSP, SWP, MIRA. Furthermore, this study investigates the impact of Q-learning method against the routing algorithms (e.g. MHA, WSP, SWP, MIRA) on multimedia QoS traffic under a realistic environment in terms of PSNR, Throughput, Packet Loss, Delay and flow rejection. By this, the decision drives the network to meet the requirement in SLA agreement by allocating the right routing in the network life. The main contribution of this work are as follows:

- RL-based Network Routing Management: The proposed framework is incorporated to employ statistical decision-making methods for traffic distribution. By this, the network management becomes self-organizing in applying an adaptive bandwidth provision scheme on learning basis from the experiments of past trails. The inclusion of AI in SDN-based network shall increase the cognitive capabilities of the decision-making procedure. Based on the learning concept, the proposed approach shall choose the best action from a set that optimizes the network performance. As a result of that, policy management system can improve its decision in order to reduce the risk of policy violation.

- Intelligent Network Resource Allocation: Resources in the network are planned to be dynamically allocated depend directly on the network state. The aim is to intelligently distribute the bandwidth over the network in order to cope with the continuous changes in the network. With the help of intelligent decision-making process, the network shall dynamically adjust its state depends on the business needs.

Nowadays diverse traffic classes (such as video and data) are transferred across the communication networks. Due to the limitation of network resources, the quality of the traffic flows is affected as a consequence from the network congestion. This leads to several traffic problems such as packet loss, low throughput, which have a great impact on the users' perceived quality. Therefore, to guarantee certain requirements, traffic classes poss different QoS requirements. Guarantee of QoS provisioning has become an active field of research especially considering its importance for applications that require data delivery under certain QoS constraints (e.g., multimedia and voice data). The QoS-enabled networks provide significant performance improvements for QoS services by ensuring sufficient bandwidth, controlling latency and reducing packet loss [164–166].

Over the past years, two QoS architectures were defined by IETF: IntServ [167] and DiffServ [168]. IntServ is a flow-based with fine-grained mechanism for traffic management. Under this approach, the network resources are reserved explicitly for specific individual traffic flows while it guarantees an end-to-end QoS delivery. However, IntServ has scalability and complexity issues [169]. On the other hand, DiffServ is a class-based with coarse-grained mechanism. It works at class level, where a class is an aggregate of many flows. The specification of policies are saved at the border nodes of DiffServ domains. According to a certain level of treatment, packets belonging to a desired service are marked and classified by the border nodes of the network domain. Core nodes in the domain use packet queuing and scheduling management techniques to forward packets according to a per hop behavior and to their class priority. Border nodes in different network domains need to map the flows carefully in other domains in order to reduce the loss of performance quality. Due to the per hop behavior and flow-aggregation model by the approach, the guarantee for end-to-end QoS solution becomes unpredictable and approximate [170]. Although DiffServ solved the problem of scalability by IntServ, it presents inevitably the problem of performance loss due to the class discretization and the fixed number of preconfigured traffic classes [83]. The current QoS architectures are still not the successful QoS support for service providers, enterprises and end users [42, 169].

In this work, the research proposes a new framework based on a set of routing algorithms while integrating reinforcement learning. The approach is not focusing on designing a routing algorithm that meets multiple constraints. Instead it utilizes the reinforcement learning method to select the optimal routing algorithms that achieves the best results on the network under dynamic network conditions. It decides intelligently on the routing algorithm based on the reward that complies to the SLA requirement of service. With the help of reinforcement learning, the research proposes a framework towards autonomic system with respect to the QoS guarantee. The framework satisfies the properties of autonomic system with self-configuration, self-healing, self optimization [158, 159].

4.4.1 Framework Architecture

Figure 4.8 illustrates the proposed RL-based framework built on top of the SDN architecture, that consists of:

(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.

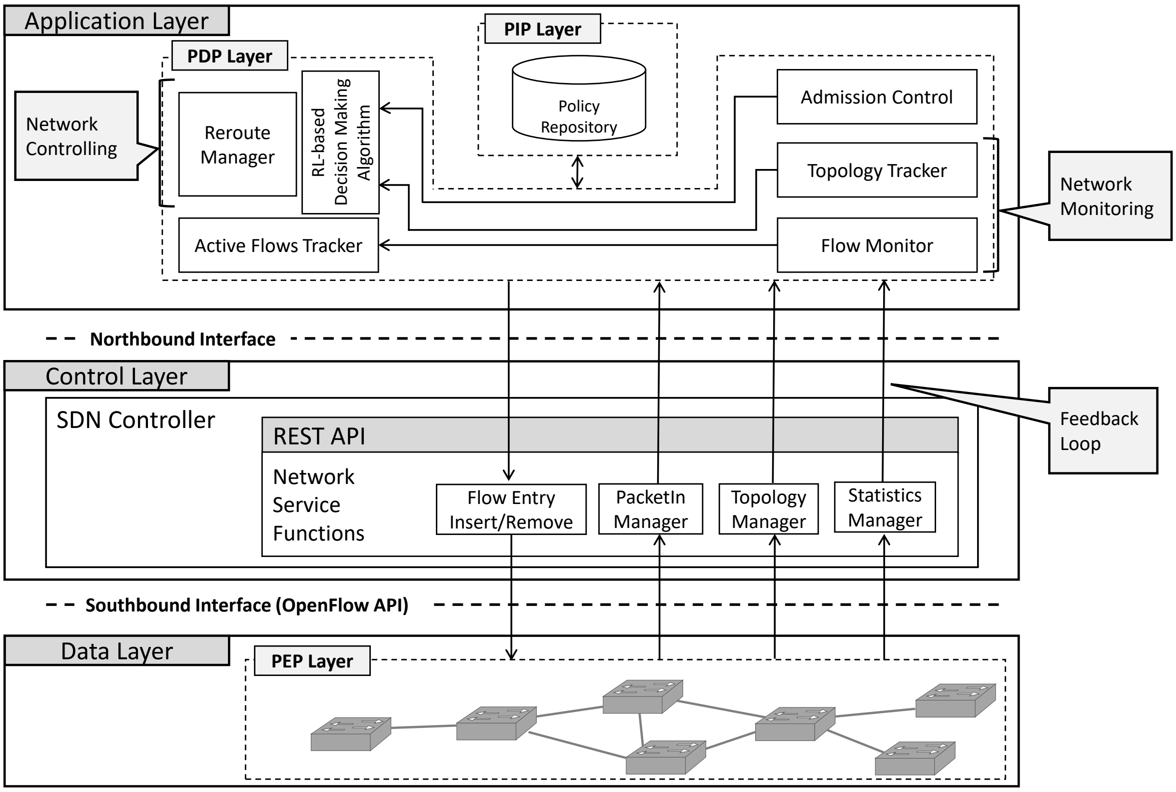

Figure 4.8: Proposed PBNM-based SDN Framework

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 [45]; (2) Shortest Widest Path (SWP) finds the feasible path with the maximum available bandwidth among the set of existing routes [45]. 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 [45]. 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 [46].

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.

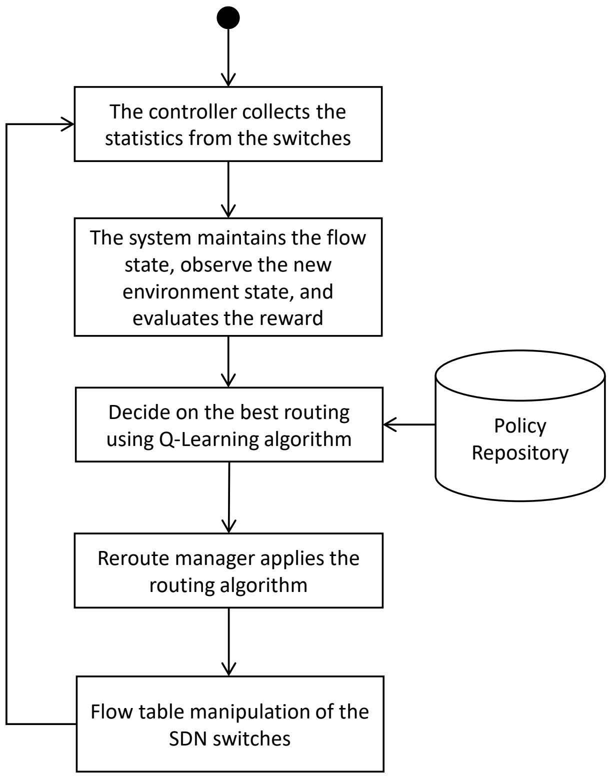

Figure 4.9: Network management work flow

4.4.2 Network Management Function

The functional components of the proposed architecture are mapped to the general three-level PBNM framework (PIP layer, PDP layer, and PEP layer) [41]. Two cases are identified for managing the network state: (1) *upon receiving a new route request* - Initially, the controller receives a packet-in message belongs to the newly incoming request. On behalf, the admission control in the application layer decides whether the request is accepted or rejected based on the resource availability. This is realized by building a reduced network graph while eliminating all links that has residual bandwidth less than the requested one. If the source and destination node are not connected in the reduced graph, then the request is rejected. On the other hand if the request is accepted, the service type is identified at first and the group of routing algorithm belonging to the service type (e.g. QoS) is applied. By this, the routing algorithm finds a feasible path where the links have residual bandwidth equal or greater than the demanded bandwidth. (2) *upon monitoring the network state* - Figure 4.9 shows the work flow of this case. Initially, the flow monitor component collects periodically the flow statistics from the network. On behalf, 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. For this, it invokes the reroute manager to apply the current routing algorithm on the actual active flows in the network.

4.4.3 System Model

Figure 4.10 illustrates the proposed system model that contributes to making decisions based on the RL algorithm through interactions with the SDN-based network environment. The system is composed of the RL-based decision making algorithm that aims at finding the most suitable routing algorithm that maximizes the long-term reward from the environment. While the routing control block performs the rerouting tasks by utilizing the knowledge of network topology and active flows. Here the environment is defined as a network of certain topology size ψ with the flows $f \in (F_{qos} \cup F_{bkg})$ transmitting through it, while the RL-based algorithm senses the network by measuring the throughout $\tilde{a}_{f_v}$, packet loss $\tilde{b}_{f_v}$ and rejection rates $\tilde{c}_v$ of the actual flows f in the network.

Let a network topology of certain size γ contains a set of possible paths P that connects the source and destination nodes. Based on the knowledge of the network topology and the active flows, the routing algorithm routes the flow f_v of a certain class v on one path $p \in P$ based on the action o_{qos} decided by the RL-based decision making algorithm. By this, the routing control block updates the flow table of the SDN switches to reroute the actual flows accordingly. While using the reward feedback R_v through an interaction with a dynamic environment, the RL-based algorithm can train itself progressively, from state to state, such as: in a certain current state $s(t)$, an action o_{qos} is decided from a set of routing algorithms $O_{qos} = \{MHA,\ WSP,\ SWP,\ MIRA\}$ and applied; when the SDN system moves into the next state $s(t+1)$, the reward function evaluates the system performance and updates the value of the action selection in the previous state $s(t)$. In order to satisfy the objective, each flow f_v is assigned to an indicator set $\{x_f, y_f, z_f\}$ that determines if the flows respect the requirement set $q_{v,thr}, q_{v,loss}, q_{v,rej} \in Q_f$. Based on the reward value, the Q-learning algorithm updates the Q-table based on the following equation [20]:

$$Q(s(t), o_{qos}(t)) = Q(s(t), o_{qos}(t)) + \alpha \Big\{ R + \lambda max_{o_{qos}} Q(s(t+1), o_{qos}(t)) - Q[s(t), o_{qos}(t)] \Big\} \tag{4.6}$$

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 $max_{o_{qos}} Q(s(t+1), o_{qos}(t))$ 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, λ and α represent the discount factor and the learning rate respectively, with values between 0 and 1.

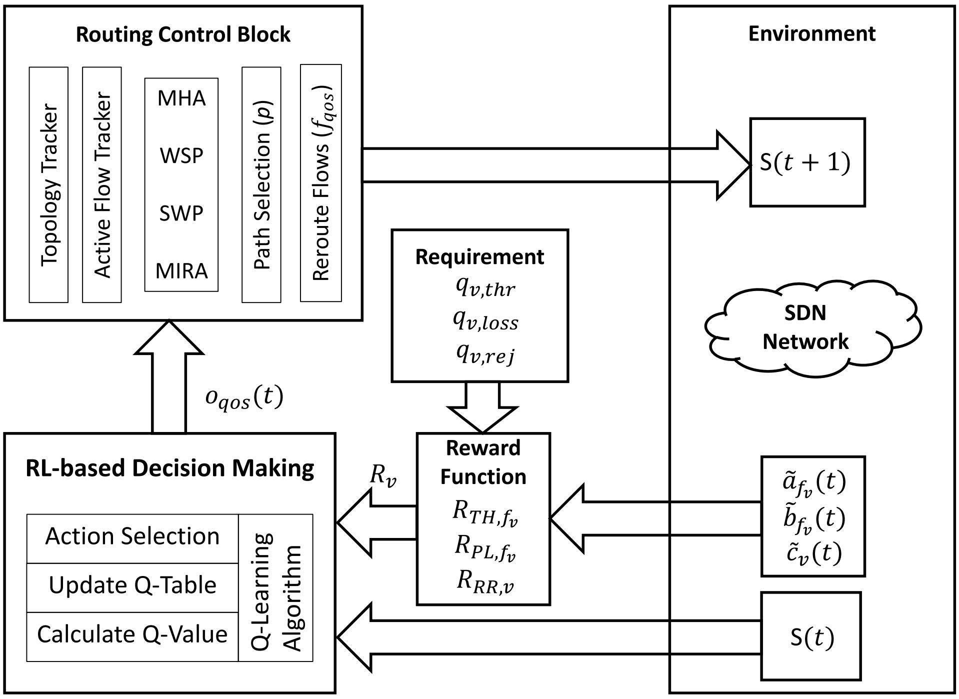

Figure 4.10: The proposed system model

4.4.4 Design of the Learning Framework

This section introduces the problem formulation and describes the optimization problem with the design of RL-based solution.

4.4.4.1 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.

Table 4.4: 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
γ	topology size
ψ	traffic load
β_{qos}	The throughput requirement is met by all QoS flows
α_{qos}	The packet loss requirement is met by all QoS flows
ϕ_{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

However, each flow f that belongs to a certain flow 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 4.4 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:

$$\sum_{p\in P} u_{p,f} = 1, \quad \forall f \in F, \tag{4.7}$$

$$u_{p,f} \in \{0,1\}, \quad \forall f \in F, \forall p \in P, \tag{4.8}$$

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 (4.9) and (4.10) 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.

$$\sum_{f\in F} d_{l,f} \cdot a_f \leq C_l, \quad \forall l \in E, \tag{4.9}$$

$$d_{l,f} \in \{0,1\}, \quad \forall f \in F, \forall l \in E, \tag{4.10}$$

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 [171]. Here, constraints (4.11)-(4.16), and 4.16 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 $\{F_{qos}, F_{bkg}\}$.

$$\sum x_f = 0, \quad \forall f \in F, \tag{4.11}$$

$$\sum y_f = 0, \quad \forall f \in F, \tag{4.12}$$

$$\sum z_f = 0, \quad \forall f \in F, \tag{4.13}$$

$$x_f \in \{0,1\}, \quad \forall f \in F, \tag{4.14}$$

$$y_f \in \{0,1\}, \quad \forall f \in F, \tag{4.15}$$

$$z_f \in \{0,1\}, \quad \forall f \in F \tag{4.16}$$

The main objective is to route the flows in a network to maximize the flows that satisfy the SLA requirement in terms of throughput, packet loss and rejection rate. However the optimization problem is subject to constraints that needs to be satisfied to solve the problem as below:

$$\text{maximize} \sum_{f \in F} \sum_{p \in P} u_{p,f}(x_f \cdot y_f \cdot z_f), \tag{4.17}$$

$$\text{subject to (4.7)-(4.16).} \tag{4.18}$$

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 [172]. Moreover, the generalization of decision-making given by RL is more flexible [173, 174].

4.4.4.2 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 [175]. In this research, RL is used to solve the optimization problem in (4.17) and (12) given the RL ability to deal with objective maximization problems [48]. 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 (4.17) with the research problem defined in 4.4.4.1, 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 [48]. 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 [176]. Next, we introduce the state and action spaces, as well as the proposed reward function used to model the proposed decision-making problem.

4.4.4.3 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:

$$S = [\gamma,\ \psi,\ \beta_{qos},\ \alpha_{qos},\ \phi_{qos}]\,, \tag{4.19}$$

where $\gamma \in \{scale_{small},\ scale_{medium},\ scale_{large}\}$ is the topology size, while the size of traffic load is denoted by $\psi \in \{load_{low},\ load_{medium},\ load_{high}\}$. Parameter β_{qos} indicates if the throughput requirement is met for the particular QoS service type. Similarly, the state parameter α_{qos} indicates if the packet loss rate requirement of QoS service type is met. Finally, ϕ_{qos} shows if the rejection ratio is satisfying a certain level. All these parameters have a binary representation calculated as follows:

$$\beta_{qos} = \begin{cases} 1 & \text{if } \sum x_{f_{qos}} = 0, \\ \\ 0 & \text{if } \sum x_{f_{qos}} > 0, \end{cases} \tag{4.20}$$

$$\alpha_{qos} = \begin{cases} 1 & \text{if } \sum y_{f_{qos}} = 0, \\ \\ 0 & \text{if } \sum y_{f_{qos}} > 0, \end{cases} \tag{4.21}$$

$$\phi_{qos} = \begin{cases} 1 & \text{if } \sum z_{f_{qos}} = 0, \\ \\ 0 & \text{if } \sum z_{f_{qos}} > 0. \end{cases} \tag{4.22}$$

4.4.4.4 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.

4.4.4.5 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:

$$R_{TH,f_v} = \begin{cases} 1 - \left[\frac{q_{v,thr} - \tilde{a}_{f_v}}{q_{v,thr}}\right] & if \quad \tilde{a}_{f_v} \leq q_{v,thr} \\ \\ 1 & if \quad \tilde{a}_{f_v} > q_{v,thr} \end{cases} \tag{4.23}$$

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:

$$R_{PL,f_v} = \begin{cases} 1 - \left[\frac{\tilde{b}_{f_v} - q_{v,loss}}{\tilde{b}_{f_v}}\right] & if \quad \tilde{b}_{f_v} \geq q_{v,loss} \\ \\ 1 & if \quad \tilde{b}_{f_v} < q_{v,loss} \end{cases} \tag{4.24}$$

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:

$$R_{RR,v} = \begin{cases} 1 - \left[\frac{\tilde{c}_v - q_{v,rej}}{\tilde{c}_v}\right] & if \quad \tilde{c}_v \geq q_{v,rej} \\ \\ 1 & if \quad \tilde{c}_v < q_{v,rej} \end{cases} \tag{4.25}$$

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:

$$R_v = w_{TH} * \frac{\sum_{f_v \in F_v} R_{TH,f_v}}{N} + w_{PL} * \frac{\sum_{f_v \in F_v} R_{PL,f_v}}{N} + w_{RR} * R_{RR,v} \tag{4.26}$$

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:

$$R = \underbrace{w_{HD\_Video} * R_{HD\_Video}}_{QoS\ service\ type} + \underbrace{w_{SD\_Video} * R_{SD\_Video} + w_{ftp} * R_{FTP} + w_{http} * R_{HTTP}}_{Background\ service\ type} \tag{4.27}$$

The weights are assigned based on the traffic ratios in the setup. The ratios are provided by Cisco [177] as described later in Section 8.1. 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%.

4.5 Chapter Summary

This chapter introduces the overall system architecture of the proposed framework and provides details of the three main contributions of this research: (1) measurement collection and probabilistic-based routing (2) Policy-based QoS Management under SDN, and (3) RL-based Decision Making for Routing Algorithms under Policy-based SDN Environment.

Monitoring in SDN-based network is considered a vital engine to realize autonomous and intelligent networks. Three main monitoring techniques are introduced: a) The first method is based on the continuous update where flow statistics data are either pushed to the controller or requested from the controller periodically; b) To reduce the monitoring overhead, the link state update is introduced to impose the push of state only when certain constraints are fulfilled; c) The third method is based on the compression technique and it extends the existing functionality of the SDN switch. The compression-based monitoring solution is proposed to decrease the monitoring overhead on the control link between forwarding and control layers, while increasing the observability of network state.

However, to overcome the problems introduced by the inconsistent update state of the switches, a probabilistic-based routing solution is also proposed, referred to as *BaProbSDN*. *BaProbSDN* makes use of the probability distribution information to find the most probable feasible path that has the best chance to satisfy the QoS constraints. Furthermore, to improve the QoS provisioning over SDN the integration of PBNM into the proposed framework is introduced. By using a loop chain approach between network monitoring and policy validation/enforcement, the framework can achieve end-to-end QoS. Upon detection of a policy violation, the proposed framework implements two flow management techniques: *rerouting* and *rate limiting*. Finally, reinforcement learning algorithm for routing management under SDN environment is proposed. The proposed solutions are evaluated through simulations and experimental testing as described in the following chapters.

Chapter 5

Measurement Collection and Probabilistic-based Routing

This chapter presents the performance evaluation of the first contribution of this research, namely: measurement collection and probabilistic-based routing*. The chapter presents the details of the experimental setup environment, the scenarios and case studies used for the performance evaluation and provides a comprehensive discussion on the results obtained*

5.1 Compression-based Monitoring Technique

5.1.1 Simulation Setup Environment

In order to evaluate the the proposed sparsity approximation algorithm for SDN monitoring applications (as described in 4.2.1.3), the experimental setup deployed in this study is illustrated in Fig. 5.1. The test-bed consists of three main elements: (i) Mininet [178] - used to emulate the SDN switching data plane; (ii) external Floodlight OpenFlow controller [53] - provides RESTful API and network services like the flow entry update; and (iii) the application layer - containing the routing and log management for performance evaluation. The Floodlight SDN controller is widely used in the research community especially because it is user friendly and the implementation of new services is easy. Furthermore, it is used in this research because it supports the necessary functionality for the study requirement such as the query of flow statistics [179]. The log management collects the statistical data for performance comparison. To evaluate the proposed method Iperf [180] tool is used to generate the network traffic between the hosts. Iperf tool is originally used to measure network and bandwidth performance [181]. A linear network topology of two hosts and four SDN OpenFlow 1.3 switches is used to evaluate the proposed method. The link speed is set to 50 Mbps in order to avoid congestion in the network links. By this, the data set include the dynamic changes of the flow patterns. For the test, the data set of a single switch is used to validate the solution.

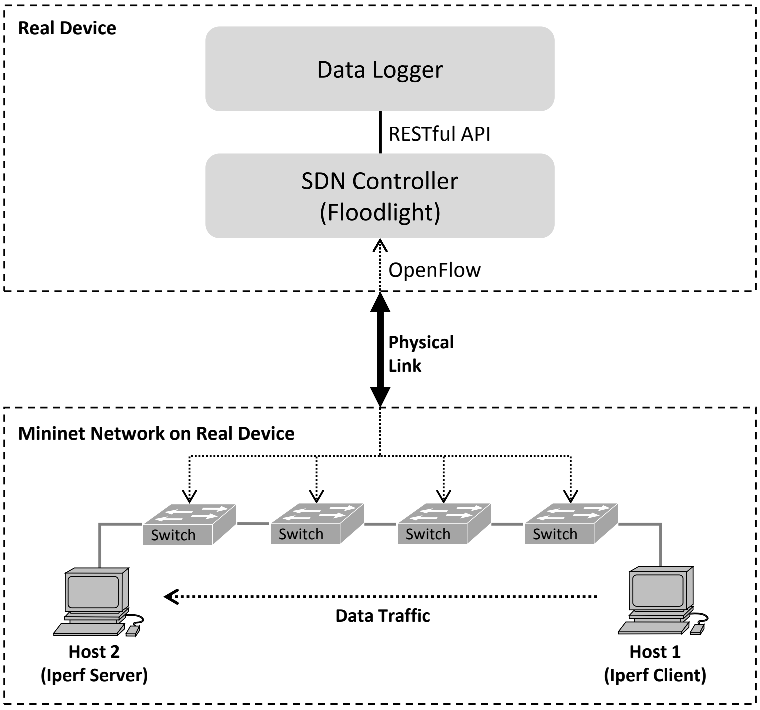

Figure 5.1: Testbed for experiment evaluation

5.1.2 Network Emulation Scenarios

A definite number of UDP sessions are distributed uniformly over the entire test period of 72 hours with a new UDP flow created every 25sec up to a total of 10000 UDP sessions. In the test a UDP protocol is used for traffic generation in order to avoid TCP's hand-shake and data retransmission and by this a fixed transmitted amount of data is guaranteed. The data rate per session is distributed uniformly between 0.1 and 1.0Mbps, while the session duration is distributed exponentially with a mean of $1/\mu$ seconds [182]. In the data logger (see Fig. 5.1), the statistical request and query between the control and data plane are stored at every second. The data are captured and sampled frequently in order to evaluate the solution using Matlab tool.

The following performance metrics are used to quantify the reliability of the proposed algorithm:

- **Signal-to-Noise Ratio (SNR):** It represents the ratio between the signal power to the noise power contained in the signal [183]. To validate the operation of technique, the parameter indicates the level of noise introduced by the concept in the recovered signal. SNR value calculated as follows:

$$SNR = 10 \cdot log_{10} \left(\frac{P_s}{P_N} \right) [dB] \tag{5.1}$$

where P_s is the power of the recovered signal and P_N is the power of the noise.

- **PSNR:** It is the ratio between the peak value of the signal with respect to the noise level contained in the signal. The calculation is based on section 2.1.2. In order to compute the SNR and PSNR in this chapter, the real signal is represented by the original measured throughput data, while the noise is the error introduced by the compression process.

- **Normalized Mean Absolute Error (NMAE):** It is the normalized value of the mean of the absolute errors over the average of original signal [184]. In this chapter, the absolute error is the absolute value of the difference between the recovered signal built by the reconstruction method and the original signal. It indicates a measure of the noise introduced by the algorithm and it is calculated as follows:

$$NMAE = \left(\frac{\frac{1}{n}\sum_{i=1}^{n} |r_t - s_t|}{\frac{1}{n}\sum_{i=1}^{n} |s_t|} \right) \tag{5.2}$$

where $|r_t - s_t|$ is the absolute error between the recovered signal r and the original signal s.

- **Cross Correlation (CC):** It is a measure of similarity between the input and output signal of the system [183]. In this chapter It used to find whether a phase shift is introduced between the original and recovered signals. For discrete-time signal, it is calculated as follows:

$$CC = \sum_{n=-\infty}^{\infty} x[k] \cdot y[n-k] \tag{5.3}$$

where x is the input or original signal and y is the output or recovered signal.

5.1.3 Network Emulation Results and Analysis

This section presents the performance evaluation of the proposed sparsity approximation algorithm for SDN monitoring applications (as described in 4.2.1.3). For the evaluation purpose, the data set is divided into two groups: (1) the training data for establishing the dictionary, and (2) the testing data for system performance evaluation. In general, the testing data set is selected in a lower proportion than the training set. In this study, the percentage is divided approximately 60% for the training data, while 40% for the testing data. In this study, the size of the training data set is studied to examine the impact of the training data set in the performance and accuracy of the proposed method. By this, a different size of training data within the entire training set is selected to study the accuracy of proposed sparsity approximation algorithm in terms of performance metrics like the signal noise. The data of the experiment is split into the first part is dedicated for the training purpose while the second part is made for the testing. In order to evaluate the performance of the proposed method, it was compared against another prediction algorithm based on the ARIMA model. The results were averaged over 50 independent runs of the algorithm with a confidence interval of 95%. The experimental setup is used to generate the data traffic over a period of 72 hours with a sampling rate of 1sec which allows us to undertake the evaluation on a large data set. The amount of data exchanged between the control and data planes is controlled through the Compression Ratio (CR). The CR defines the level of compression performed on the data according to the formula $CR = s/M$ where s is the sparsity level which is the number of non-zero components in the sparse vector x and M is the raw data set length which is equal to the length of dictionary raw. In order to evaluate the proposed compression technique, different sparsity levels indicated by CR are examined. CR has direct impact on the overhead introduced on the control link between the two planes. In this work, the goal is to decrease the overhead by achieving the highest possible compressible data, while maintaining a high accuracy of the recovered signal at controller. The reconstruction accuracy is studied from two perspectives: the size of the training data $|T|$ used for the learning process and the size of the dictionary $|D|$. The training data $|T|$ is selected in the range of 2000 to 38000 throughput elements with an interval of 6000 elements. The aim is to find the appropriate size of $|T|$ and $|D|$ that bring considerable performance benefits. For the optimal network operation, it is desired to use a relatively low amount of data for training as the sparsity approximation algorithm starts after the dictionary $\hat{D}$ is estimated. On the other hand, the performance of different dictionary sizes is also investigated. The size condition for sparse approximation over an over-complete dictionary is satisfied when $K > M$. For this purpose, the dictionary size is incremented according to $K = M + f \cdot M$ where f is set to $\{0.5, 0.75, 1.0\}$ and this leads to the corresponding size of $\{1350, 1560, 1800\}$ respectively.

Figure 5.2 shows that for the training data size of 2000 and 8000, there is a gain in the SNR and PSNR of about 2.2dB. However the impact on SNR and PSNR reduces when the training data size goes above 8000. For example, for $|D| = 1560$ the SNR increased from 32.8dB to 32.9dB when the training data increased from 14000 to 32000 respectively. The main reason is that the dictionary reaches a state where even with additional training data there is no further improvement achieved. Similarly, Fig. 5.3 shows that NMAE decrease, while CC is not effected by the size of $|D|$ and $|T|$. In contrast, the increase in the dictionary size $|D|$ has slight impact on the recovered signal inaccuracy.

Figure 5.3 also shows that a dictionary size and training data size of 2000 are not desirable as the training data is not large enough to capture the optimal shape. Thus, in this work the training data size is set to 8000 and the dictionary size of 1560 is further used for the performance evaluation.

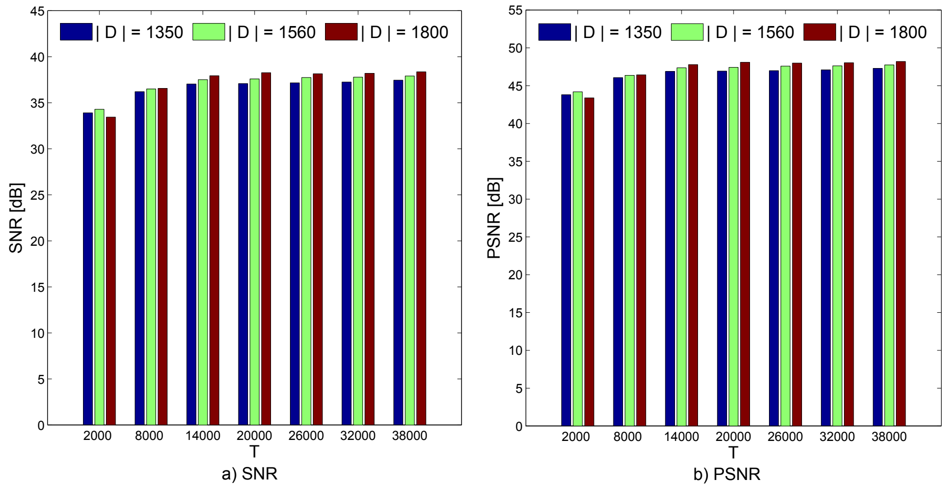

Figure 5.2: Comparison of performance metrics SNR and PSNR with different training data size $|T|$ and dictionary size $|D|$, while sparsity level $s = 10$

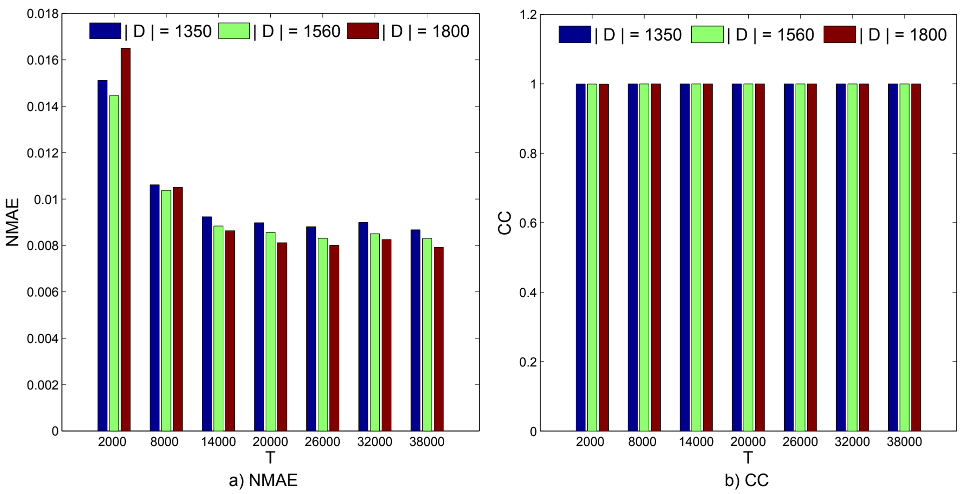

Figure 5.3: Comparison of performance metrics CC and NMAE with different training data size $|T|$ and dictionary size $|D|$, while sparsity level $s = 10$

Table 5.1 shows the trade-off between the sparsity level and the quality of the recovered signal in terms of PSNR, SNR, NMAE and CC. A high SNR and PSNR with a low NMAE value means a better recovery quality due to the small error introduced to the recovered signal. It can be seen that the inaccuracy between the original and the recovered data decreases as CR increases. This is because for a lower sparsity level the number of dictionary basis vectors used for recovery is smaller than for higher sparsity level.

Table 5.1: Performance metrics for the experimental setup under different sparsity levels, for $|T| = 8000$ and $|D| = 1560$

CR	PSNR [dB]	SNR [dB]	NMAE	CC
0.067	36.353	26.117	0.03835	0.9968
0.167	42.697	32.741	0.01650	0.9993
0.333	46.089	36.269	0.01063	0.9997
0.500	51.369	41.538	0.00586	0.9999
0.667	57.493	47.683	0.00278	1.0000
0.833	63.562	53.752	0.00146	1.0000
1.000	73.214	63.408	0.00048	1.0000

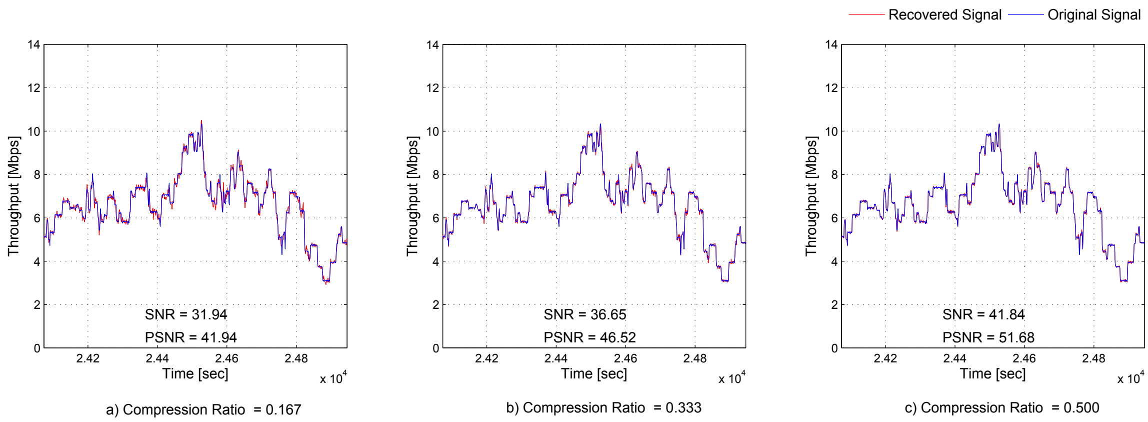

Figure 5.4: Comparison of original with recovered signal under different CR, while $|T| = 8000$, and $|D| = 1560$

To find the CR that leads to good performance measures, the strength level of the original data is taken as a reference level to define an absolute power value. Overall, the experimental results show that for the case where $|D| = 1560$ and CR $= 0.333$ the SNR value of 36.269 dB indicates a recovered signal with power of -0.00095 dB to the reference level and NMAE value of 0.01063. This means that the errors do not have a significant impact.

In order to evaluate further the performance of the proposed system, real Internet traffic traces from Center for Applied Internet Data Analysis (CAIDA) were employed [185]. CAIDA collects traces on several backbone links and it provides a big data set of recorded traffics. For the study, four one-hour long data sets are employed and two data sets are used for training and the other two are used for the system evaluation. Table 5.2 shows the comparison of sparsity level for CAIDA traces. The results show that the performance with real traces is close to that of the experimental results.

Table 5.3 shows the total communication amount in kilo-bytes for the original OpenFlow messages exchanged without compression and the proposed compression technique with flow aggregation. The

Table 5.2: Performance metrics for CAIDA data set under different sparsity levels, for $|T| = 8000$ and $|D| = 1560$

CR	PSNR [dB]	SNR [dB]	NMAE	CC
0.067	25.496	16.860	0.11699	0.9747
0.167	34.047	26.558	0.03823	0.9970
0.333	45.798	38.772	0.00921	0.9998
0.500	54.446	47.464	0.00339	1.0000
0.667	62.766	55.809	0.00126	1.0000
0.833	68.554	61.598	0.00066	1.0000
1.000	68.729	61.776	0.00064	1.0000

Table 5.3: Total Overhead Communication per Switch [KB]

CR	Without Compression	With Compression
0.067	16623.047	144.066
0.167	16623.047	185.624
0.333	16623.047	254.887
0.500	16623.047	324.149
0.667	16623.047	393.412
0.833	16623.047	461.435
1.000	16623.047	505.214

results show that the compression method reduces significantly the communication overhead in the control plane. For example, for CR = 0.333 the proposed method reduces the overhead with up to 98% when compared to the case without compression. Figure 5.4 shows the quality of the sparse approximation and its signal recovery in the time domain. As expected, the results show that the recovery accuracy is increasing as CR increases. Although Fig. 5.4 is subject to higher CR value and lower SNR value, the recovered signal captures mainly the signal's profile. For this, further post-processing filter mechanisms can be introduced to improve the signal quality.

Furthermore, the proposed compression method is also compared to the case where the statistical parameters are first estimated by the switch and then sent to the controller. For the purpose of comparison, an Auto-Regressive Integrated Moving Average (ARIMA) (r, d, m) model [186] is implemented and evaluated for the same experimental setup. However, instead of carrying out the compression method, the ARIMA model estimates the statistical parameters for the Auto-Regressive term (r), Integrated term (d) and Moving-Average term (m) of a stationary and non-stationary time-series data [186]. The model is identified with ARIMA(1,1,1) by using the training data and auto-correlation function. The results of ARIMA model show, compared to Table 5.3, that it reduces the control overhead with up to 98% to 342KB when compared to the solution without compression (e.g., 16623.047KB). However, these benefits come at the cost of low SNR (e.g., 14.2dB) and NMAE of 0.1591. The proposed compression-based approach achieves further reductions in the control overhead for CR >= 0.50. Moreover, compared to the compression-based approach, for a CR of

0.50 there is up to 65% increase in SNR (e.g., 41.538dB) and NMAE of 0.00586. Furthermore, an alternative time series prediction method like the Kalman filter and Hidden Markov Model can be applied to estimate the measurement. As a prerequisite, Kalman filter requires to define the modeling of measurement and system space in order to operate within the expected output. However finding the modeling of prediction and measurement update that suits the network real-world data becomes difficult to describe mathematically. On the other hand, this research proposes another way to utilize the compression-based algorithm to exchange less statistical data from the forwarding plane to the control plane.

5.2 Probabilistic-based QoS Routing

5.2.1 Simulation Setup Environment

The performance of the proposed probabilistic routing algorithm BaProbSDN (as described in 4.2.2), is assessed through simulations using Matlab. The implementation is carried out using the generic Matlab toolbox. Matlab is widely used in the research area to run simulation and validate algorithms. In this work, the standard Matlab is used with no need of special toolbox. The entire software is created where the necessary code and functions are implemented. Additional Matlab scripts are implemented to evaluate the output signals and draw the statistical comparison images for analysis. In this work, the entire Matlab-based simulation is hosted on a Windows system (1.80GHz processor of 4 cores with memory size of 8GB). The Internet Service Provider (ISP) topology as illustrated in Figure 5.5 is implemented for performance evaluation, the topology is well known in study of routing algorithms and it represents the nationwide network of US ISP [187–189]. The number of switches in the topology is 18, while the number of links inter-connecting the switches is 30. Each switch is connected to a single host that generates the traffic flows. In the simulation. the link capacity is set to 100Mbps. As the study is executed by MATLAB simulation, the link speed was set to represent a realistic link capacity.

5.2.2 Simulation Scenarios

The network simulation employs a traffic generation model that loads the network using two kinds of request arrivals: one kind of request is triggered by each host in order to generate the best-effort traffic in the background while the other kind of request is generated by the source host (H1) to destination node (H2) for QoS traffic.

The traffic arrival follows a Poisson distribution with rate λ while the active period of a connection

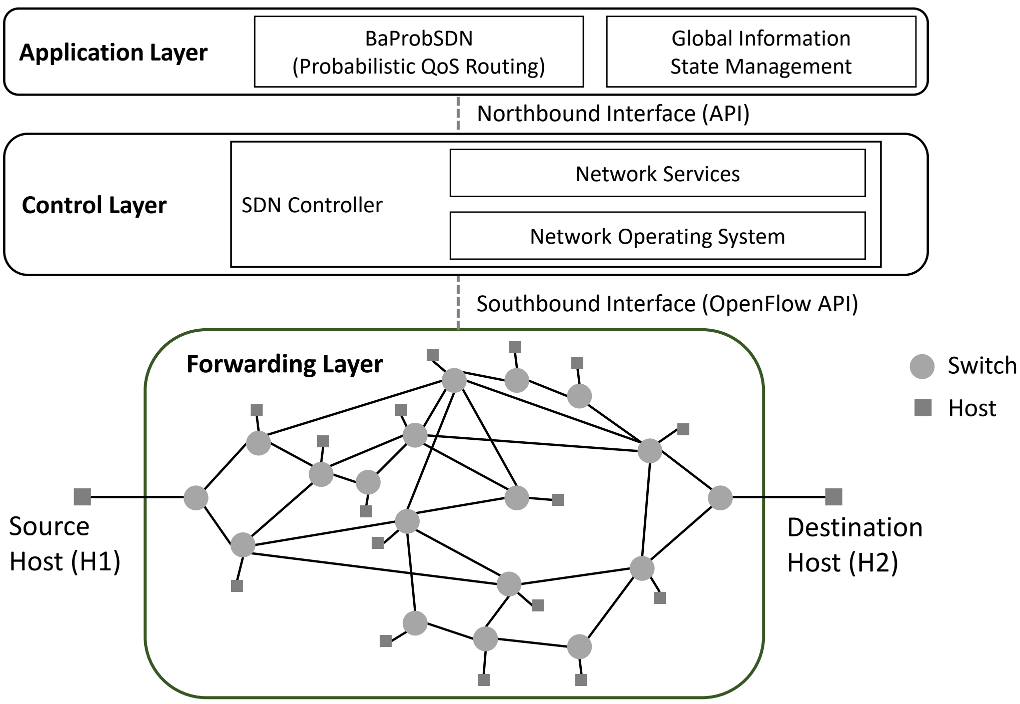

Figure 5.5: Considered ISP topology

exhibits an exponential distribution with a mean of $1/\mu$ seconds [182]. The destination node is chosen at random other than the source node. For the traffic load, the request arrival rate λ is chosen to determine the overall load. In the simulation, two traffic patterns are used to load the network, the first behaves for long duration with a mean of $\mu = 70$ seconds and small bandwidth requirements distributed uniformly between 1Mb and 10Mb, while the second indicates a short duration with a mean of $\mu = 10$ seconds and larger bandwidth requirements of uniform distribution between 20Mb and 30Mb. The two patterns were shared equally in the simulation. The links are assumed to be bi-directional with the same bandwidth capacity C. For the QoS routing traffic, the source and destination pairs are fixed during the simulation period and their requested bandwidth is uniformly distributed between 10Mb and 40Mb. Within the SDN network, every switch samples the link metric at every 1 second interval and it sends the link state information to the controller based on the link state update policy decision as described in 4.2.1.2.

In order to assess the algorithm, the performance metric of bandwidth blocking rate is employed to reflect the success of the algorithm to set an end-to-end QoS path according to the bandwidth metric. A rejected request indicates that the end-to-end QoS path contains at least one link that does not satisfy the bandwidth constraint. Therefore by measuring the rate, it shows if the algorithm generates an end-to-end QoS path P with the following rule $min_{(i,j)\in P} \lfloor b\,(i,j) \rfloor > b_{req}$. The bandwidth blocking rate is defined as in eq. 5.4 [190].

$$bandwidth\ blocking\ rate = bbr = \frac{\sum_i^N b_{rej}(i)}{\sum_j^M b_{req}(j)} \tag{5.4}$$

where b_{rej} is the bandwidth of the rejected request and N is the number of rejected requests, while b_{req} is the requested bandwidth, $\sum_j^M b_{req}(j)$ is the sum of the bandwidth of the arrived requests and M is the number of arrived requests.

5.2.3 Simulation Results and Analysis

The performance of the proposed BaProbSDN algorithm (as described in 4.2.2) is evaluated through simulations using Matlab and it is compared against WSP [191] as described in 2.4. The WSP algorithm selects the route with the largest amount of available bandwidth, while the proposed probability-based algorithm finds the path that is most likely to satisfy the bandwidth availability. Therefore the WSP routing algorithm is chosen to be compared with the proposed algorithm in terms of finding the path that satisfies the bandwidth availability constraint.

This section presents the performance comparison of the two algorithms under various settings of the link state update policy. The results were averaged over 50 simulation runs and they were obtained with a 95% confidence interval. The 95% confidence interval is a range of values calculated from the data set that, most likely, includes the true value of what is estimated about the population [192]. Similar definition, it is the 95 percent confidence and certain that the estimate lies within the range of the upper and lower values specified by the confidence interval [193]. The lower and upper limit of the confidence interval is computed in Matlab by using the formula $ci = \mu \pm z \cdot \sigma/\sqrt{n}$ where μ is the mean and σ is the standard deviation, while z is the z-value and it sets to 1.96 for 95% confidence interval and n is the number of samples. As SDN separates the control plane from the data plane, the control and information messages are exchanged frequently between the switches and the controller for reasons like checking of resource availability or capacity planning. In particular, for routing purposes, a significant amount of overhead is often introduced to keep the network state at the controller, as accurate as possible especially when the network size becomes larger. Therefore, it is essential to quantify the algorithm performance and the overhead introduced due to frequent update messages. The aim is to minimize the overhead associated with the routing in the SDN network while ensuring that the QoS guarantees are satisfied. To this extent, the threshold-based triggering policy together with the HDT are introduced. Figure 5.6 illustrates the state information update rate under various threshold and HDT values. It can be noticed that the introduction of threshold-based link update policy reduces the number of advertised states in the network. For example, for the threshold value of 0.5 and HDT=0s there is 84.05% decrease in the state information update rate. It can be seen that the variation of update percentage decreases with the increasing threshold and HDT values. For a threshold value of 0.3 and HDT=20s the update rate reaches 3.78%.

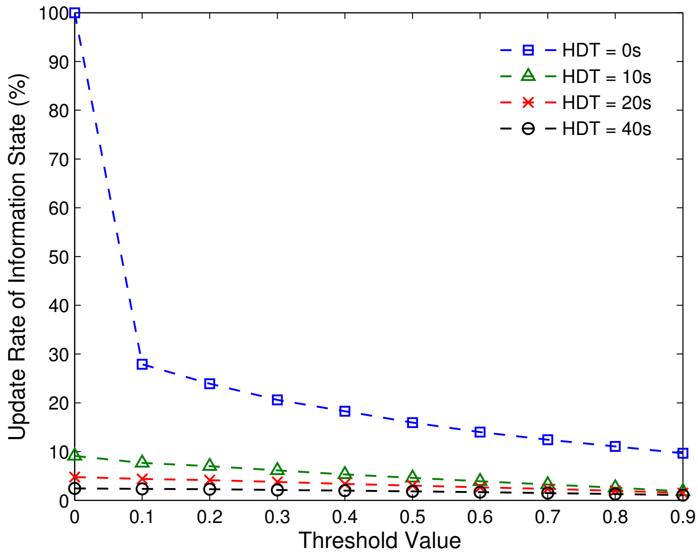

Figure 5.6: State information update rate for different HDT and threshold-based link update policy

As previously mentioned, the BaProbSDN method computes the link probability on a set of observations taken over a WS. Therefore, Fig. 5.7 shows how the WS impacts the performance of BaProbSDN method. The result shows that the blocking rate becomes steady as WS increases. For example, for HDT = 10s and threshold value of 0.3 BaProbSDN achieves 26.90% in the bandwidth blocking rate when WS = 100. Thus for the rest of simulation, the WS is set to 100 seconds.

Figure 5.8 illustrates a comparison of the algorithms under varying threshold values. It can be seen that BaProbSDN reduces the bandwidth blocking rate when compared to the WSP algorithm. For example, for Thr=0.6, BaProbSDN can achieve up to 5.50% decrease in the bandwidth blocking rate when compared to WSP. Moreover, WSP has a close performance to the BaProbSDN method because the group of links that support enough bandwidth depend on the requested bandwidth lying in the range of $b_{req} \notin [b_{k-1}(1+thr), b_{k-1}(1-thr)]$. WSP will present an improved performance in the presence of imprecision only if those links that contribute to the QoS path do not lie in this range.

Figure 5.9, presents the results in terms of bandwidth blocking rate, obtained when HDT value was varied between from 5 to 40 seconds. The results show that as HDT becomes larger the performance of the two algorithms decreases noticeably. It can be seen that BaProbSDN outperforms the WSP algorithms. For example, for HDT=15s and Thr=0.5, BaProbSDN can achieve up to 9.43% decrease in the bandwidth blocking rate when compared to WSP.

In order to study the impact of the traffic load, the traffic arrival rate λ was varied while other

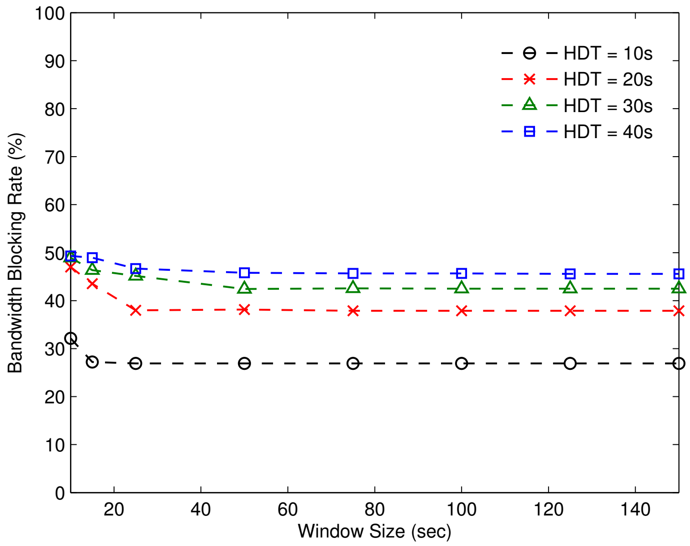

Figure 5.7: Bandwidth blocking rate of BaProbSDN method under different WS (Threshold=0.3)

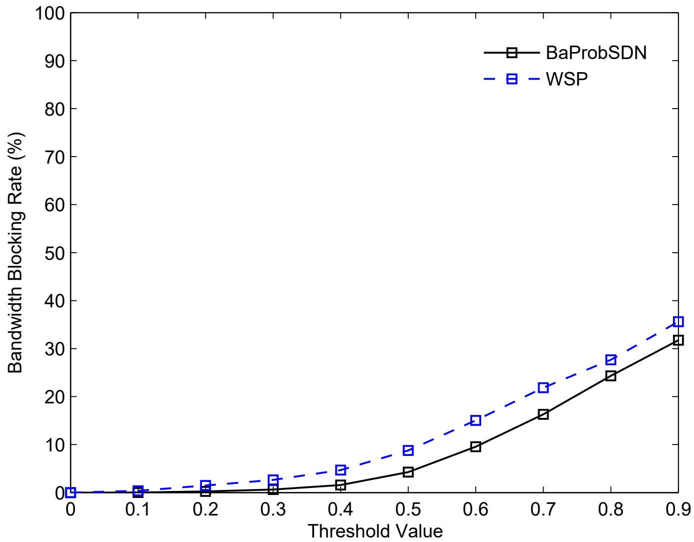

Figure 5.8: Bandwidth blocking rate for different threshold values and HDT=0s

simulation parameters were kept fixed. Figure 5.10 shows the performance of the two algorithms as a function of the network traffic load. Due to the aggregated cross traffic the link will be more exploited as the traffic load increases. It can be seen that under highly loaded network, the blocking rate of the two algorithms increases. However, when compared to WSP, BaProbSDN achieves up to 7.96% decrease in the bandwidth blocking rate for a traffic load of 140Mbps.

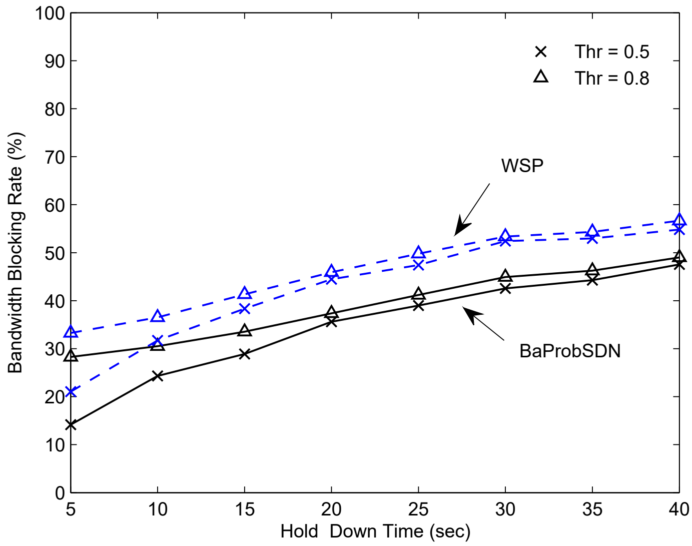

Figure 5.9: Bandwidth blocking rate for different HDT values

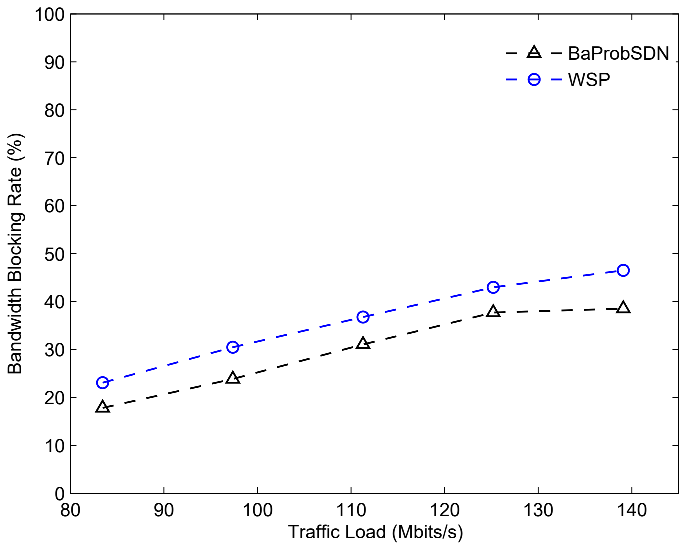

Figure 5.10: Bandwidth blocking rate for different traffic loads (Threshold=0.3, HDT=10s)

5.3 Chapter Summary

This chapter illustrates the experimental setup and the performance evaluation of the proposed solutions under the first contribution of this research, measurement collection and probabilistic-based routing. Network experimental setups and simulation scenarios are performed to evaluate the monitoring and probabilistic routing solutions. The data collection through compression technique has shown further reduction in the control link usage for network applications while increasing the network observability. The results show that the proposed method reduces the control link overhead cost with up to 98% when compared to the case of periodic acquisition network monitoring of the SDN network. The solution was analysed over a range of sparsity levels, showing that it reduces the overhead significantly while the controller recovers the sparse data with an SNR of 36.269dB and an NMAE of 0.01063 for a CR = 0.333. On the other hand, BaProbSDN makes use of Bayes' theorem and Bayesian network model to determine the link probability in terms of bandwidth availability. The proposed algorithm was compared against the WSP algorithm. The results demonstrate that BaProbSDN can achieve up to 8.02% decrease in the bandwidth blocking rate when compared to WSP For example, it achieves up to 7.41% decrease in the bandwidth blocking rate, for a threshold value of 0.5, HDT=10s, while a reduction of 95.38% is achieved in the control messages overhead.

This chapter demonstrates how the concept of compression and probabilistic routing can be used to bring benefits to SDN-based environments. In order to realize the application service, the application, control and forwarding layers are involved in the adaptation. While novel, this solution may face major difficulties in implementation on the forwarding layer. In order to do this, the realization of measurement collection through compression technique requires major updates on the OpenFlow switch design which may not be easily adopted. For example, the data in the sparsity form needs to be advertised to the application layer as described in 4.2.1.3. The data of interest is not supported by the OpenFlow protocol. As a consequence, this has an impact on the Openflow specification. On the other hand, the approach of compression needs to be trained on regular basis. Additionally, the concept needs at the final stage to be deployed on a commodity device. At the hardware level, the OpenFlow switch possess limited memory resource [194]. This added another computational and resource burden of how the training process is adopted to overcome this limitation.

In the following chapter the second contribution of this research on the integration of policy-based QoS management within SDN is evaluated and the performance of the rerouting and rate limiting methods for QoS provisioning is demonstrated. The monitoring utilizes the existing features of the switch complaint to OpenFlow specification and the approach is mainly adopting the control and application layer for the research purpose, while no extra modification on the switch level is performed. Consequently, the proposed framework is based on the approach of continuous measurement update 4.2.1.1 and utilizes the standard OpenFlow specification.

Chapter 6

Policy-based QoS Management

This chapter presents the setup of experimental environment used for testing and illustrates the results of the scenarios and case studies used for the performance evaluation of the second research contribution: policy-based QoS management.

6.1 Emulation Setup Environment

The proposed PBNM-based SDN framework (as described in 4.3) was implemented and tested under the experimental setup illustrated in Figure 6.1. The testbed consists of three main elements: (i) Mininet [178] - used to emulate the SDN data plane; (ii) external Floodlight OpenFlow controller [53] - provides RESTful API and network services like the flow entry update; and (iii) the PBNM application layer (described in Section 4.3) - containing the decision making for QoS policy configurations.

For the network testing, two kinds of testing experimental setup are generally existing for the validation of proposed solution. The test can be carried out either by network emulation-based setup or network simulation-based setup. The network emulation setup mimics virtually the realistic environment by replicating the real network setup on a single machine and it can be deployed easily to prototyping the real network. While the network simulation demonstrates the behaviour of a network by modelling the network [195]. Here, an emulation experiment setup is employed to validate the proposed algorithm under various scenarios. Mininet is a network emulator which is utilized to emulate a customized SDN-based network with OpenFlow compatible switch [178]. The tool enables researcher to run fast prototyping and experimental evaluation at no cost. The SDN controller and the entire PBNM application run on a computer and they are connected via a physical Ethernet link to other computer hosting Mininet. Ofsoftswitch13 and Dpctl [196] are used as a OpenFlow 1.3

compatible user-space software switch. The Ofsoftswitch13 is implemented in user space with Linux and the code is originally based on Ericsson Traffic Lab 1.1. One of the primary application of the switch is the support of per-flow meter to rate limit the packets.

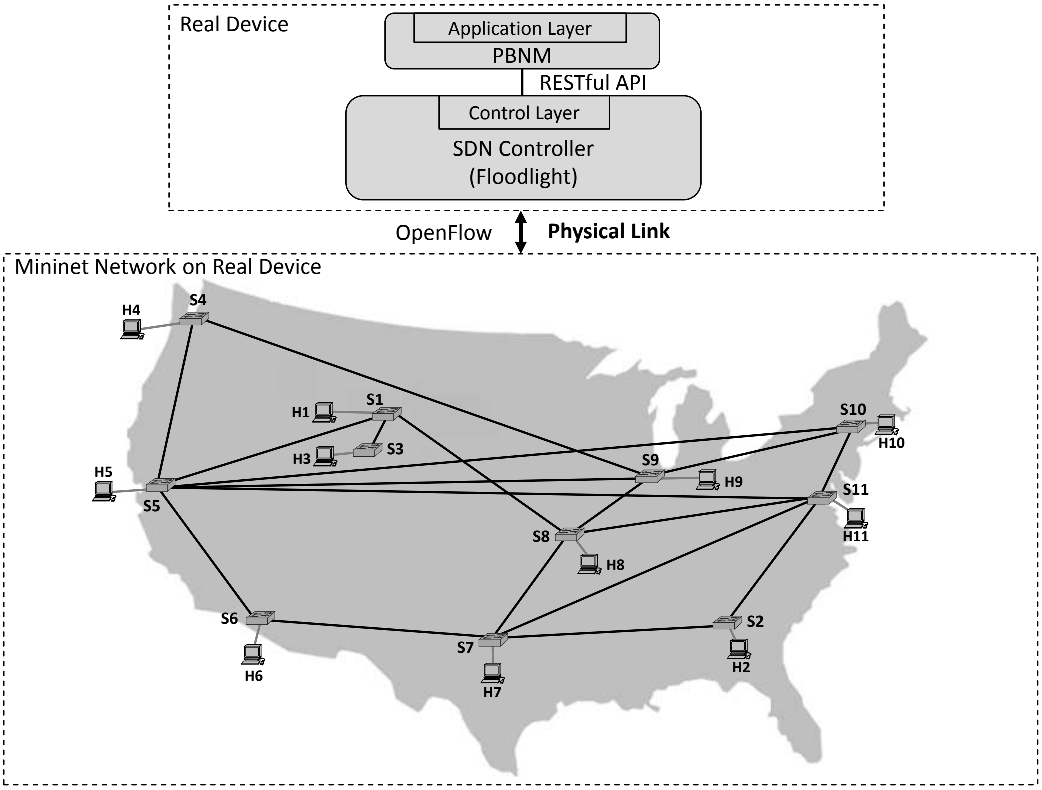

Figure 6.1: Experimental Setup using the Sprint network topology

In order to evaluate the approach, a realistic ISP topology is used. The Sprint IP backbone and customer topology as depicted in Figure 6.1 were used for the experimental setup, with the network nodes being replaced by SDN-Openflow enabled switches. The topology was taken from Internet zoo topology [197] and consists of 11 nodes that are interconnected through 18 connectivity links. Each switch has a host directly connected that generates traffic flows. The entire network topology is treated as a one SDN domain with a single global SDN controller. Consequently, the experimental scenario is simplified to prevent complexity and problems arising by integration of multiple controllers such as efficient communication process, load balancing [198, 199]. On the other hand because of the processing capacity limitations in the experimental setup, each link in the topology operates at the rate of 1 Mb/s. A larger link capacity in the topology engages a higher number of HTTP and FTP flows in order to sustain the traffic mix ratio provided by Cisco. Thus, larger number of flows requires higher processing time to accomplish adequate task by individual flow such as the flow statistics query or rerouting.

Table 6.1: Parameters of traffic modeling and setup

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

Video streaming traffic is generated using the VLC player, while background traffic like HTTP is generated using Ostinato [200], a network traffic generator tool. In this way, it is possible to evaluate different traffic mix and load on the network. The traffic generated within the experimental setup consists of: guaranteed QoS traffic such as video streaming and best-effort traffic represented by a mix of video and web flows used as background traffic. Table 6.1 illustrates the parameters used for video traffic, the experiment duration as well as the traffic mix. The traffic mix ratio is determined based on the statistics provided by Cisco [201] such that 80% of the total traffic is represented by video traffic and the remaining 20% is represented by HTTP traffic. The parameters for the HTTP traffic model [2–4] used are listed in Table 6.2. The HTTP traffic is modeled as ON/OFF period, where the ON period corresponding to the transmission time and the OFF period corresponding to the packet inter-arrival time. For each traffic request, the source and destination host pairs are selected randomly following a uniform distribution.

Table 6.2: Model parameters of web traffic [2–4]

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

6.2 Emulation Scenarios

In order to evaluate the proposed PBNM-based SDN framework under dynamic network conditions and policy violations, a scenario with a mix of QoS and best-effort flows is considered. The proposed

PBNM framework integrates two methods that could be triggered to overcome the policy violation, namely the rerouting and rate limiting. For the purpose of performance evaluation, the QoS policy rule is defined for this scenario as: the QoS video traffic from source Host 2 (H2), as indicated in Fig. 6.1, directed to the destination Host 4 (H4) has a minimum bandwidth threshold of 600 Kb/s and the maximum threshold for packet loss rate is set to 2%. The characteristics of the QoS video traffic are listed in Table 6.1. The host pair (H2 to H4) was selected to represent the longest distance within the network to increase the likelihood of disturbing the QoS video flow by other background traffic. The distance between two nodes representing here the minimum number of hops between them. In order to disturb the QoS video flow, a mix of video and HTTP traffic as background traffic are generated between random hosts maintaining the 80% to 20% ratio, with the traffic characterized according to [201].

6.3 Emulation Results and Analysis

For the performance evaluation, the performance of the proposed PBNM-based SDN framework is compared against the default configuration of the SDN-based network without the PBNM framework. The default SDN is a plain SDN controller without PBNM capability and it maintains a minimal routing logic based on finding the shortest path when a new traffic request arrives. The comparison is performed on the same random seed to reproduce a deterministic trail. Each experiment is repeated three times and the average outcomes are evaluated. Both approaches, such as rerouting and rate limiting of the proposed PBNM-based SDN framework are considered. The performance evaluation is done in terms of Throughput, Packet Loss Rate, Latency, PSNR, SSIM and MOS of the QoS video flow as defined in 2.1.

6.3.1 PBNM-based SDN Framework with Rerouting

In this setup the proposed PBNM-based SDN framework has the rerouting module enabled. Thus, when QoS policy violation is detected the framework reroutes the disturbing traffic and gives priority to the QoS video flow. As a first step in the route setup phase, the route manager selects the least loaded path (S2-S11-S5-S4) for the QoS video traffic between H2 and H4. Figure 6.2 illustrates the throughput, packet loss rate and latency measurements for the QoS video flow under PBNM-based SDN framework with rerouting and the default SDN without PBNM. It can be noticed that three policy violations were detected by the framework.

The results show how the policy condition on the shared link S11-S5 is being strictly violated for the first violation. During the experimental run, the monitoring component identifies at time-stamp

64 that the packet loss rate exceeded the limit of 2% imposed by the QoS policy rule. The packet loss rate is caused due to the shared resources on the common link which becomes congested. Due to this, the violation detector identifies the best-effort flow from Host 11 (H11) to Host 5 (H5) as a disturbing flow and it routes it on an alternative path S11-S10-S5. In order to determine the sharing link causing the problem, the violation detector uses the supervised neural network to check if the given link is involved. As a consequence, the violation detector releases the event of policy constraint breaching and notifies the route manager. Other violations are identified in the time-stamp 132 and 252.

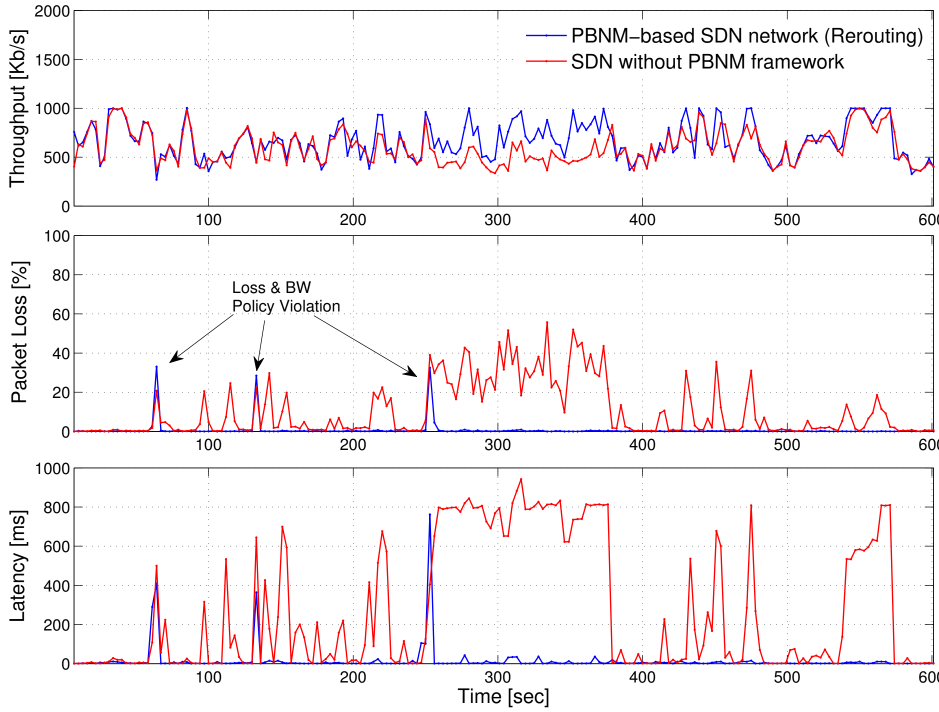

Figure 6.2: Throughput, packet loss rate, and latency of QoS traffic flow for PBNM-based SDN framework with rerouting and default SDN without PBNM

Table 6.3 lists the average PSNR and SSIM for the QoS video flow as well as the mapping to the MOS done according to Table 2.1, of both the proposed PBNM-based SDN framework with rerouting and the default SDN. The MOS value is a five point scale as illustrated in section 2.1.2. The results indicate that when using the PBNM-based SDN with rerouting, the user perceives the video quality as *Excellent* based on both PSNR and SSIM to MOS mapping. Whereas in the case of default SDN, the user perceived quality for the QoS video flow is *Poor* (based on PSNR to MOS mapping) towards *Fair* (based on SSIM to MOS mapping). Thus, by using the proposed PBNM-based SDN framework with rerouting there is an increase of 94% in PSNR as compared to the default SDN.

Figure 6.3 illustrates a comparison snapshot of the QoS video frame from the original transmit-

Table 6.3: Average PSNR to MOS and SSIM to MOS Mapping

	Performance Metrics			
	PSNR	MOS	SSIM	MOS
PBNM with rerouting	46.61	5 (Excellent)	0.99	5 (Excellent)
Default SDN	23.97	2 (Poor)	0.94	3 (Fair)

ted video, the video frame received after the proposed PBNM-based SDN framework performed the rerouting and the video frame as received using the default SDN. It can be noticed that the QoS video frame quality becomes noticeably poorer relative to the original video frame when the default SDN framework is used with a PSNR of 15.39dB indicating a *Bad* user perceived quality according to the MOS mapping in Table 2.1. However, by enabling the proposed PBNM-based SDN framework with rerouting the quality of the video frame improves considerably, with a PSNR of 50.52dB representing *Excellent* user perceived quality.

Figure 6.3: Quantitative video frame quality comparison: a) original image, b) proposed PBNM-based SDN framework with rerouting (PSNR = 50.52dB, MOS = 5 - Excellent), and c) default SDN (PSNR = 15.39dB, MOS = 1 - Bad)

6.3.2 PBNM-based SDN Framework with Rate Limiting

In this setup the proposed PBNM-based SDN framework has the rate limiting module enabled. When a QoS policy violation is detected, the rate limiting module will throttle the output rate of the background best-effort traffic by dropping packets while the traffic flows maintain the same route. This is done, in order to ensure an end-to-end QoS guarantee for the video flow and to control the high throughput aggregates in the network.

Figure 6.4 illustrates the throughput, packet loss rate and latency measurements for the QoS video flow under PBNM-based SDN framework with rate limiting and the default SDN without PBNM. In this case, the metering manager takes a rate limiting measure to resolve the misbehavior of the background best-effort traffic flows. The results show that if no network adjustment would be considered (e.g., default SDN-based network without PBNM), the QoS video flow throughput would continue to suffer from the impact of packet loss and delay.

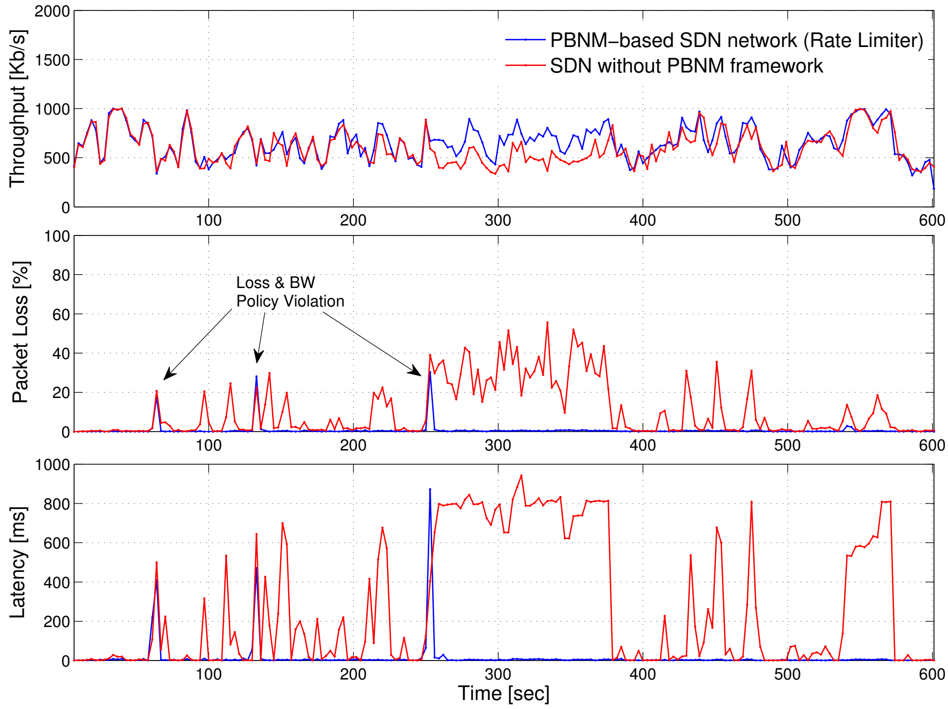

Figure 6.4: Throughput, packet loss rate, and latency of QoS traffic flow for PBNM-based SDN framework with rate limiting and default SDN without PBNM

Table 6.4 lists the average PSNR and SSIM for the QoS video flow as well as the mapping to the MOS done according to Table 2.1, of both the proposed PBNM-based SDN framework with rate limiting and the default SDN. The results are similar to the case where the PBNM-based SDN framework with the rerouting approach is used. It is observed that both methods led to user perceived quality improvements when compared to the default SDN approach. Results show that the proposed PBNM-based SDN framework with rate limiting can achieve up to 91% increase in PSNR with a *Excellent* user perceived quality compared to the default SDN where the user perceived quality is *Poor* (based on PSNR to MOS mapping) towards *Fair* (based on SSIM to MOS mapping).

Table 6.4: Average PSNR to MOS and SSIM to MOS Mapping

	Performance Metrics			
	PSNR	MOS	SSIM	MOS
PBNM with rate limiting	45.81	4 (Excellent)	0.99	5 (Excellent)
Default SDN	23.97	2 (Poor)	0.94	3 (Fair)

Figure 6.5 illustrates a comparison snapshot of the QoS video frame from the original transmitted video, the video frame received after the proposed PBNM-based SDN framework performed the rate limiting and the video frame as received using the default SDN. Similarly to the previous rerouting setup, it can be noticed that the QoS video frame quality is significantly improved by using the proposed PBNM-based SDN framework with rate limiting from *Bad* quality as perceived with the default SDN to *Excellent* quality.

Figure 6.5: Quantitative video frame quality comparison: a) original image, b) proposed PBNM-based SDN framework with rate limiting (PSNR = 50.45dB, MOS = 5 - Excellent), and c) default SDN (PSNR = 15.39dB, MOS = 1 - Bad)

6.3.3 Monitoring Overhead vs. Application Performance

This section analyzes the trade-off between the monitoring overhead introduced and the application performance. For this purpose, several experimental runs are conducted using the same setup but with different monitoring update intervals, such as 3, 6 and 9 seconds. The choice of the monitoring update interval values, starting from 3 seconds above is done because the SDN controller from the experimental setup needs time to perceive a consistent image of the entire network and to take the necessary measures in order to avoid the aftermath of policy violation. The results are listed in Table 6.5 for the default SDN and the proposed PBNM-based SDN framework with rerouting and with rate limiting. The results indicate that as the monitoring update interval increases the application performance decreases. This is because the SDN controller will take longer to detect and respond to the misbehaving best-effort traffic that affects the quality of the QoS video flow. However, even with the increased monitoring update interval both methods of the proposed PBNM-based SDN framework outperform the default SDN. For example, for an monitoring update interval of 9 seconds the quality of the QoS video flow is still perceived as *Good* (based on PSNR to MOS mapping) towards *Excellent* (based on SSIM to MOS mapping) for both proposed approaches, rerouting and rate limiting compared to *Poor* (based on PSNR to MOS mapping) towards *Fair* (based on SSIM to

MOS mapping) as perceived when the default SDN framework is used.

Table 6.5: Averaged performance evaluation for different monitoring update intervals (3, 6, and 9 seconds)

Performance Metrics	Default SDN	PBNM with rerouting			PBNM with rate limiting		
		3	6	9	3	6	9
Throughput [Kb/s]	605	645	651	648	621	616	630
Packet Loss [%]	10.22	0.65	1.02	1.35	0.69	0.97	1.38
Latency [ms]	268.67	14.87	13.06	12.15	14.04	13.60	12.43
PSNR [dB]	23.97	46.61	45.13	43.46	45.81	44.47	43.22
MOS (PSNR)	2 (Poor)	5 (Exc.)	5 (Exc.)	4 (Good)	5 (Exc.)	4 (Good)	4 (Good)
SSIM	0.94	0.99	0.99	0.99	0.99	0.99	0.99
MOS (SSIM)	3 (Fair)	5 (Exc.)	5 (Exc.)	5 (Exc.)	5 (Exc.)	5 (Exc.)	5 (Exc.)

Figure 6.6 shows the overall amount of monitoring overhead introduced on the control path and for different monitoring update intervals for the proposed PBNM-based SDN framework regardless of the approach being used, such as rerouting or rate limiting. Messages of type $OFPT\_STATS\_REQUEST$ and $OFPT\_STATS\_REPLY$ are used for measuring the throughput and packet loss, while the latency measurement is based on injecting $OFPT\_PACKET\_OUT$ messages as probe packets into the network and waiting for receiving $OFPT\_PACKET\_IN$ messages from the controller. The results show that the monitoring overhead is inversely proportional to the update interval. For example, the communication overhead is reduced by up to 63% when the update interval changes from 3 to 9 seconds. However, this comes at the cost of twice the packet loss rate and 10% decrease in PSNR. Thus, the trade-off between the introduced overhead and the application performance needs to be considered.

Although the introduction of PBNM scheme in SDN network adds more network overhead than the default SDN, the results show that the performance of the QoS application is significantly improved.

6.4 Chapter Summary

This chapter presented the performance evaluation of the second contribution of this research, *a policy-based network management framework over SDN*. Upon detection of a policy violation two flow management techniques are implemented, such as: rerouting and rate limiting. The proposed framework was implemented and evaluated within an experimental testbed setup. The results indicate that the proposed PBNM-based SDN framework enables QoS provisioning and outperforms the default SDN in terms of throughput, packet loss rate and latency. For example, the proposed PBNM-based SDN framework for rerouting can achieve up to 94% increase in the average PSNR when compared to the default SDN, increasing the user perceived quality from Poor to Excellent. On the other hand,

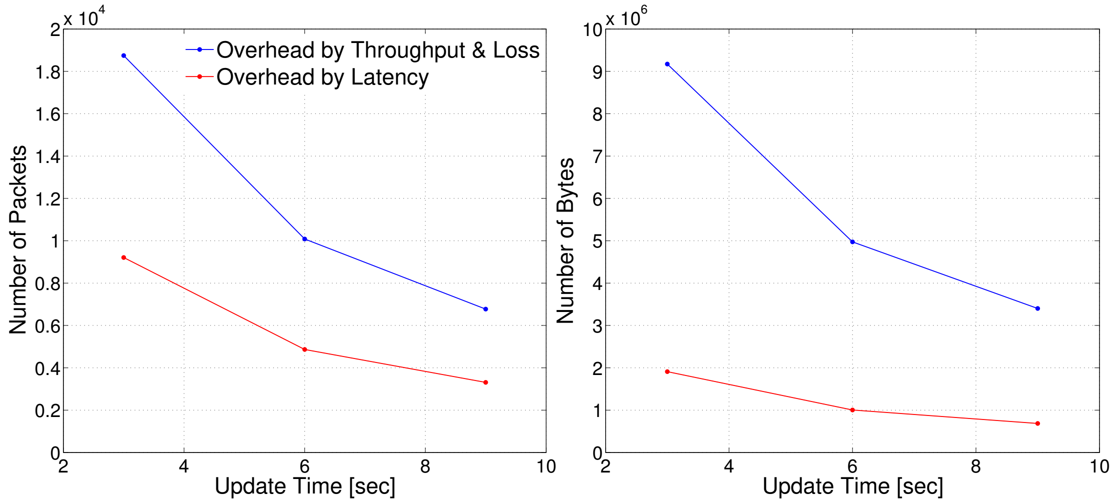

Figure 6.6: Monitoring Overhead for different update intervals (3, 6, and 9 seconds)

the proposed PBNM-based SDN framework, like rerouting and rate limiting demonstrate comparable performance.

Based on the approach discussed in this chapter a comprehensive study on the performance evaluation of various state-of-the-art routing algorithms under SDN is introduced. Consequently, the work in the following chapter will extend the experimental setup introduced here to accommodate a more realistic and dynamic network environment for evaluating a group of state-of-the-art routing algorithms. While the results of the performance evaluation in this chapter demonstrated that routing can be employed to manage the QoS provisioning under SDN network, the next chapter will investigate this further for the ultimate goal of allocating routing algorithms intelligently within the framework by employing machine learning.

Chapter 7

Performance Evaluation of Routing Strategies over SDN

This chapter presents a study that compares state-of-the-art routing algorithms over multimedia-based SDN networks considering a more realistic environment with dynamic network conditions and various topologies. Routing algorithms remain a key element of the networking landscape as they determine the path the data packets are to follow. This study investigates the impact of state-of-the-art centralized routing algorithms (e.g. MHA, WSP, SWP, MIRA) on multimedia QoS traffic under a realistic environment. The performance evaluation is carried out under an experimental setup environment and is done in terms of PSNR, Throughput, Packet Loss, Delay and QoS rejection.

7.1 Experimental Setup Environment

This section presents the experimental setup, and the evaluation scenarios considered. The experimental setup deployed in this study is illustrated in Fig. 7.1. The test-bed consists of three main elements: (i) Mininet [178] - used to emulate the SDN data plane; (ii) external Floodlight OpenFlow controller [53] - provides RESTful API and network services like the flow entry update; and (iii) the application layer - containing the routing and log management for performance evaluation. The log management collects the statistical data for performance comparison, while the route management represents the network element that makes routing decisions based on the pool of algorithms.

In the work of this chapter, the entire experiment is hosted on a powerful machine to accommodate the traffic load. The SDN controller and the entire routing management application run on a virtual computer (2.2GHz multiprocessor of 4 CPU units with memory size of 16GB), while the Mininet test-bench is running on another virtual machine (2.2GHz multiprocessor of 4 CPU units with memory size of 32GB). Each virtual machine is running Linux-Ubuntu Server. Open vSwitch [202] is used as a software SDN switch (see Fig. 7.1). 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.

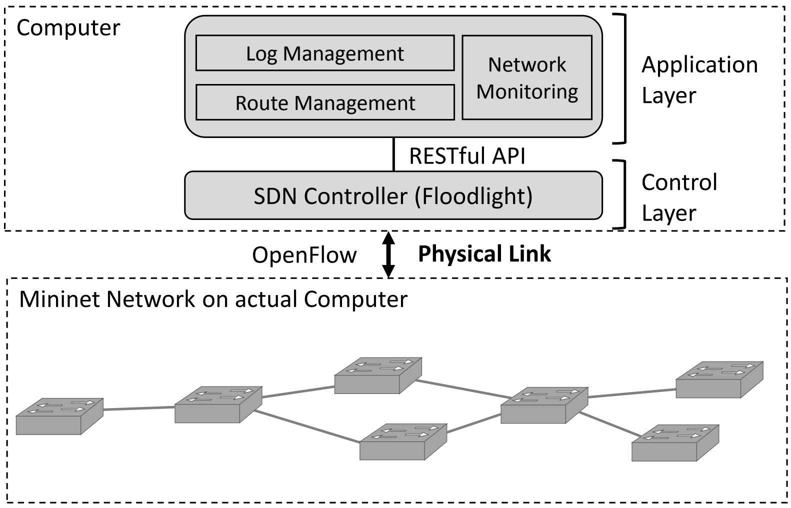

Figure 7.1: Experimental Setup for Performance Testing (The shown topology here is an example for illustration purpose only)

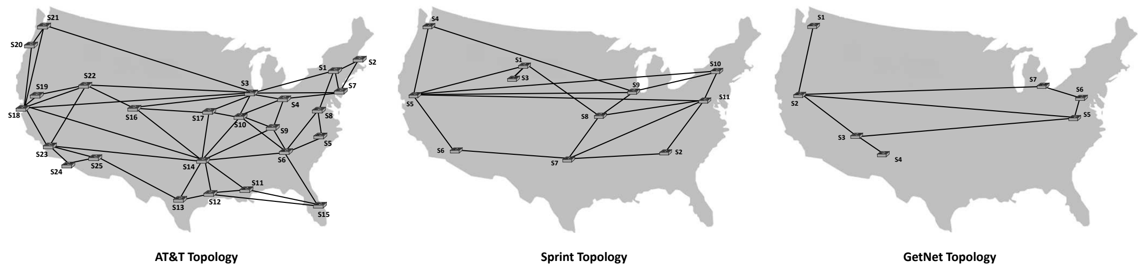

Figure 7.2: Experimental Setup using the following network topologies: AT&T (large-scale topology), Sprint (middle-scale topology), and GetNet (small-scale topology)

The performance evaluation of the routing algorithms is performed under dynamic network conditions and over three realistic network topologies selected from Internet Topology Zoo as illustrated in Fig. 7.2:

- 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.

Video streaming traffic with a Variable-Bit-Rate (VBR) encoder is generated using the VLC player, while HTTP and FTP traffic are generated using Ostinato [200] traffic generator tool. In this way, it is possible to evaluate different traffic mix and load on the network. The traffic generated within the experimental setup consists of: guaranteed QoS traffic such as the video streaming and best-effort traffic represented by video, web, and file transfer flows used as background traffic. The traffic mix ratio is determined based on the statistics provided by Cisco [201] such that 80% of the total traffic is represented by video traffic at year 2020 and the remaining 20% is represented by other traffic such as HTTP and FTP. In this work the ratio is calculated based on the amount of traffic generated within the network. The same ratio is maintained for different topologies and under different traffic loads.

The HTTP and FTP are built on a client-server model architecture. However, the two communications have differences. An HTTP session is used to access websites based on a sequence of request-response transactions. A web browser represents a client side that sends a request message to the web server. Upon receiving the request, the server sends the web page data. Web page traffic is usually based on smaller amount of data than the FTP traffic. On the other hand, the FTP traffic is used to transfer files over the network incorporating in general larger data to transfer.

The parameters for the HTTP traffic model [2, 3] used are listed in Table 7.1. The HTTP traffic is modeled as ON/OFF period, where the ON period corresponding to the transmission time and the OFF period corresponding to the packet inter-arrival time. For each traffic request, the source and destination host pairs are selected randomly following a uniform distribution. On the other hand, the parameters for the FTP traffic model [3] used are listed in Table 7.2. With each call, the session is used to transfer a file of a random generated size.

Table 7.1: Model parameters of web traffic

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

Several experimental scenarios are considered to validate the performance of the routing algorithms over the multimedia-based SDN environment. The aim is to study the impact of various routing

Table 7.2: Model parameters of ftp traffic

Parameters	Best-fit Distribution	Mean & Std. Deviation
File size	Truncated Lognormal	Mean = 2048 KBytes Std. dev. = 739.328 KBytes
Reading time	Exponential	Mean = 180 sec

algorithms on multimedia QoS traffic within SDN under a realistic environment with dynamic network conditions such as different topology, traffic patterns and traffic load. The performance evaluation is done in terms of throughput, packet loss, delay, QoS rejection and user perceived QoE for video streaming, using PSNR [35].

7.2 Network Emulation Scenarios

To evaluate the routing algorithms under dynamic network conditions, a scenario with a mix of QoS and best-effort flows is considered. In the performance evaluation, the following parameters are considered in order to drive a dynamic network evaluation:

- Network topology: Three different size of 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 [197].
- Traffic type: multiple QoS traffic flows are mixed with background traffic. For the guaranteed traffic, QoS-based video streaming is employed. While background traffic is represented by video, HTTP and FTP.
- Network load level: In order to evaluate the performance of routing algorithms under various network load, the number of active flows in the network at any given moment are adjusted 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 considered: 0.5 (low load), 0.75 (medium load), and 1.0 (high load). The network load NL is calculated as follows:

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

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

The entire experimental time is divided into several overlapped sessions in order to maintain a continuous traffic flow. The traffic arrival follows a uniform distribution over the duration of each session while the active period of each connection is distributed exponentially with a mean of $1/\mu$ seconds. The destination node is chosen at random other than the source node within the network. However, because of the processing capacity limitations as described in section 6.1, each link in the topology operates at the speed of 1 Mb/s.

7.3 Network Emulation Results and Analysis

Four routing algorithms MHA, WSP, SWP and MIRA as described in 2.4 were implemented in the SDN application layer. The performance evaluation of the four routing algorithms is assessed under dynamic network conditions. For each of the routing algorithms, the impact of the network topology and traffic load are studied on the multimedia QoS traffic. This section presents the evaluation results of the routing algorithms from the experimental trails. In the experimental setup shown in Fig. 7.1, the Log Management module collects the statistical traffic data (e.g. time-stamp, throughput, latency) for the performance comparison. Various performance metrics are used to assess the QoS-based video and background traffic, such as: average throughput, average packet loss, average latency, average PSNR, and the number of rejected QoS services. In this section, the average PSNR value is calculated using the PSNR values of the QoS-based videos belonging to a single scenario. The QoS-based video PSNR values are obtained by comparing the received QoS-based video with the original QoS-based video. Moreover, in order to study the impact of the traffic load on the rejection of QoS-based traffic, the results show the number of rejections for the upcoming QoS-based requests along the experiment as a function of the network traffic load. When a new request of the QoS-based services arrives, the algorithm finds a feasible path where the links have residual bandwidth equal or greater than the demanded bandwidth. In case there is no path that satisfies the bandwidth constraint, the request is rejected.

The results are summarized in tables 7.3, 7.4 and 7.5. The tables show the numerical results to demonstrate the performance differences among the four routing algorithms (MHA, WSP, SWP, and MIRA) and the impact of factors like the network load, traffic classes and topology size. The tables show the results in terms of various performance metrics for the evaluation of the QoS-based video traffic as well as background traffic such as the average throughput, packet loss, and latency. By this, it can be shown that there is generally no routing algorithm that fulfills the best expectation under all considered scenarios and networking conditions. For example, it is concluded that under the low traffic load that MHA, WSP and SWP achieve the best results in terms of minimizing the packet loss for QoS-based flows for the small scale networks, while MIRA achieves the best results for medium scale networks and SWP for large scale networks. While it is observed on the other hand that under the high traffic load, algorithms SWP and MIRA perform better than other algorithms under lower scale networks. Similarly, SWP achieves better results for medium and large scale networks like Sprint and AT&T. In the following sections the overall results are discussed in detail.

7.3.1 Impact of Network Topology

In this section, the impact of the network topology on the performance of the four algorithms is studied. As shown in Fig. 7.2, the comparison of algorithms are taken place on three different network sizes: GetNet - small scale, Sprint - medium scale, and AT&T - large scale. To simplify the comparison, this category is classified based on the type of traffic load.

7.3.1.1 Low Traffic Load

In this section the performance of routing algorithms under the low traffic is evaluated. When the size of network topology increases, MIRA and SWP algorithms achieved better results than MHA and WSP algorithms in terms of the packet loss, throughput and latency. For example, it has been observed in Tables 7.3 and 7.5 that as the size of the topology increases from GetNet to AT&T, the packet loss of quality traffic for the MHA algorithm has risen by 6.1% as compared to the MIRA with an increase of 3.6%. Similarly, the throughput of quality traffic for MHA algorithm has decreased by 9.5% while MIRA is decreased by 7.2%.

In general, the results in Fig.7.3 show that the performance of all routing algorithms decreases noticeably when the size of network topology increases. For example, when there is increase in the network size from GetNet to AT&T, the average PSNR for MHA, WSP, SWP and MIRA algorithms are decreased by 10.8, 7.7, 7, and 11.2 dB respectively. Similarly, it can be observed in Figure 7.4 that the number of rejection for quality services grows in proportion to the increase in topology size. In fact, as the size of topology increases, higher volume of flows are generated in order to achieve the same load under various topologies.

When looking at maximizing the throughput for QoS-based video flows, it has been noticed that MIRA, WSP and SWP perform the best for small scale such as GetNet. While for the medium scale network such as Sprint, MIRA outperforms other algorithms by achieving a throughput level of 508 Kb/s. For large-scale networks such as AT&T, WSP achieves better throughput for QoS-based video flows. In terms of minimizing the packet loss for QoS-based flows, MHA, WSP and SWP achieve the best results for small scale networks, while MIRA achieves the best results for medium scale networks and SWP for large scale networks. In terms of minimizing the latency, WSP outperforms the other algorithms for small scale networks, MIRA obtains the minimum latency for medium scale networks

Table 7.3: GetNet network topology: Averaged performance evaluation for routing algorithms

	Performance Metrics	MHA			WSP			SWP			MIRA		
		low	medium	high	low	medium	high	low	medium	high	low	medium	high
Quality Traffic	Throughput [Kb/s]	507	474	322	511	**484**	**345**	511	**484**	321	**512**	480	331
	Packet Loss [%]	**0.1**	**2.2**	19.1	**0.1**	2.3	18.1	**0.1**	2.3	**17.1**	1.2	2.5	17.2
	Latency [ms]	50	1178	8034	**30**	**1159**	6703	46	**1159**	**6354**	60	1163	7034
Back-ground	Throughput [Kb/s]	77	52	54	74	50	48	79	63	53	79	56	49
	Packet Loss [%]	0.2	3.7	7.2	0.2	1.5	6.4	0.2	3.5	4.7	0.5	4.1	5.5
	Latency [ms]	22	1257	2552	18	408	2147	14	857	1731	30	1400	2161

Table 7.4: Sprint network topology: Averaged performance evaluation for routing algorithms

	Performance Metrics	MHA			WSP			SWP			MIRA		
		low	medium	high	low	medium	high	low	medium	high	low	medium	high
Quality Traffic	Throughput [Kb/s]	465	403	319	477	426	411	485	453	**435**	**508**	**476**	371
	Packet Loss [%]	4.2	8.1	16.5	2.5	6.2	9.7	1.9	5.2	**6.5**	**1.1**	**2.5**	15.9
	Latency [ms]	1066	1311	1863	609	1124	1547	533	**431**	**1461**	**50**	660	1807
Back-ground	Throughput [Kb/s]	79	76	92	83	75	89	72	61	73	83	67	87
	Packet Loss [%]	0.6	2.6	4.3	0.9	1.5	3.3	0.6	0.8	3.5	0.2	1.5	2.1
	Latency [ms]	83	505	826	143	276	575	45	147	736	25	242	347

Table 7.5: AT&T network topology: Averaged performance evaluation for routing algorithms

	Performance Metrics	MHA			WSP			SWP			MIRA		
		low	medium	high	low	medium	high	low	medium	high	low	medium	high
Quality Traffic	Throughput [Kb/s]	459	428	**402**	**477**	431	385	472	430	390	475	**454**	395
	Packet Loss [%]	6.2	8.1	16.1	4.9	6.0	15.9	**4.2**	6.5	**13.9**	4.8	**5.2**	14.1
	Latency [ms]	1143	1372	1946	**1035**	1395	1654	1052	1341	**1604**	1090	**1235**	1679
Back-ground	Throughput [Kb/s]	176	87	73	180	87	66	193	81	60	159	84	95
	Packet Loss [%]	1.2	2.5	4.2	1.5	2.4	4.8	1.6	2.3	4.9	1.2	2.5	2.8
	Latency [ms]	273	425	872	240	323	574	478	613	1087	272	471	602

while for large scale networks WSP performs the best.

7.3.1.2 Medium Traffic Load

As depicted in Tables 7.3 and 7.5, the increase in the size of network topology from GetNet to AT&T showed that the packet loss for the quality traffic of MHA algorithm rises by 5.9%, while the WSP and SWP has a rise of 3.7% and 4.2% respectively. On the other hand, MIRA algorithm shows slightly better results as the packet loss of the quality traffic gets a value of 2.7%.

In addition, the user perceived quality of experience depicted in Fig. 7.3 shows a decrease for MHA, WSP, SWP and MIRA algorithms by 10.2, 11.3, 11.7, and 6.8 dB, respectively. Although the size of network topology affects the quality of videos, it is observed that the performance variation between the routing algorithms shows similar trends under the medium traffic load.

The measurement shows that, for the small scale network like GetNet, WSP and SWP algorithms perform the best while giving the highest throughput for QoS-based video flows. In medium and large scale networks such as Sprint and AT&T, MIRA achieves better throughput for QoS-based video flows. In terms of minimizing the packet loss for QoS-based flows, MHA, WSP and SWP achieves the best results for small scale networks. However, MIRA obtains better results for QoS-based video flows for medium and large scale networks such as Sprint and AT&T. In terms of minimizing the latency, WSP and SWP outperform the other algorithms for small scale networks. SWP obtains the minimum latency for medium scale networks while for large scale networks MIRA performs the best.

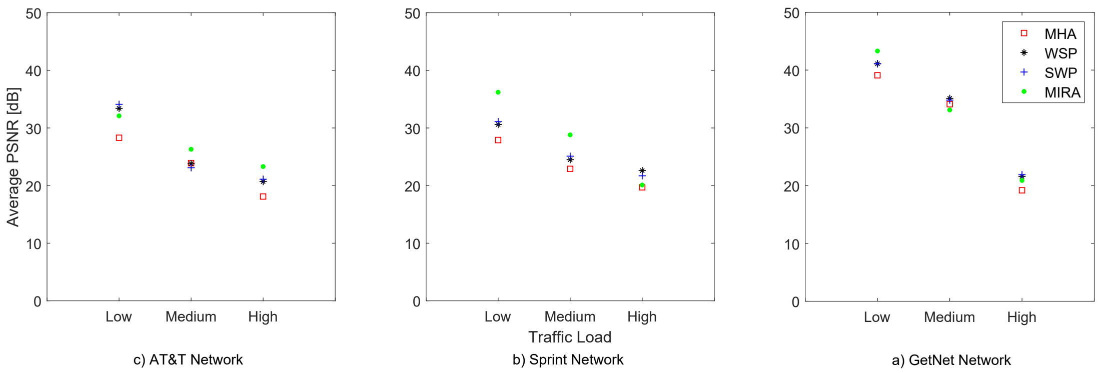

Figure 7.3: The average PSNR at different various traffic loads and network topologies

7.3.1.3 High Traffic Load

In this section, the obtained results shows different outcomes when compared to the previous sections. The routing algorithms exhibits relatively lower packet loss in larger size of network topology than in the smaller networks. For instance, the SWP algorithm shows a decrease in packet loss for quality traffic from 17.1% (GetNet) to 13.9% (AT&T). In the GetNet network the distribution of traffic are carried on smaller number of network links than in the AT&T network. Therefore it is expected to have higher traffic congestion and packet loss.

In general, it is observed in Tables 7.3, 7.4 and 7.5 that the average statistics for the background traffic show smaller values than the quality services. In fact, the background traffic contains HTTP/FTP and video traffic, while the quality services contain the QoS-based video traffic only. The HTTP/FTP traffic flows have usually much smaller load than the video traffic, hence the averaging becomes smaller for the background traffic.

To summarize the results exposed in Tables 7.3, 7.4 and 7.5, the WSP algorithm performs the best in terms of maximizing the throughput for small scale network like GetNet. For the medium scale such as Sprint, SWP provides best results. However, for large-scale networks such as AT&T, MHA achieves better throughput for QoS-based video flows. In terms of minimizing the packet loss for QoS-based flows, SWP and MIRA perform better than that of other algorithms under lower scale network. Similarly, SWP achieves better results for medium and large scale networks like Sprint and AT&T. In terms of minimizing the latency, SWP performs better when compared to other algorithms for small, medium and large scale networks.

Fig. 7.4 shows the number of rejections of the QoS-based video services. It can be seen that under highly loaded network, the rejection rate of routing algorithms increases considerably. For example, under AT&T network there is an increase of 97.2% for MHA when the load increases from low to high.

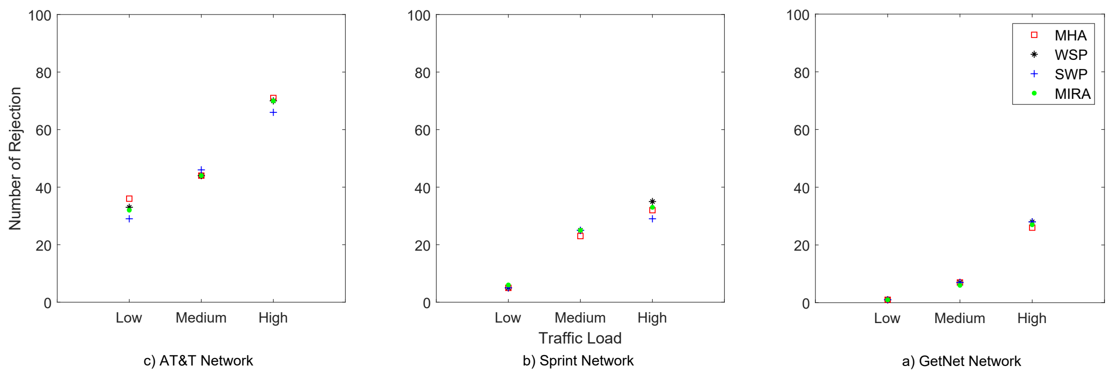

Figure 7.4: The number of rejection for quality requests at different various traffic loads and network topologies

7.3.2 Impact of Traffic load

This section presents the impact of the traffic load on the performance of the routing algorithms. The traffic load has a different impact on the performance under the same network topology. Therefore it has been classified according to the network topology.

7.3.2.1 GetNet Topology

The results depicted in Table 7.3 show that all routing algorithms reach larger packet loss when the load becomes higher. For example, MHA, WSP, SWP and MIRA algorithms gets an increase in packet loss of 19%, 18%, 17%, and 16%, respectively. On the other hand, it can be observed in the results from Fig. 7.3 that with low traffic load, MIRA algorithm performs better by reaching an average PSNR value of about 43.3dB when compared to other candidates. Under higher traffic load, SWP performs slightly better than other algorithms by reaching an average PSNR value of about 21.9dB.

While looking at the results within the same network topology but under different traffic loads, it has been noticed that under low traffic, the maximum throughput for QoS-based flows is obtained by MIRA. However, as the traffic load increases for medium traffic, WSP and SWP gets the highest throughput for the QoS-based traffic. While for high traffic load, WSP achieves the best results. In terms of minimizing the packet loss for QoS-based flows, MHA, WSP and SWP achieve the best results under low traffic, while for medium traffic load MHA performs better and SWP achieves the minimum packet loss under high traffic load. In terms of minimizing the latency for QoS-based flows, WSP performs the best for low and medium traffic while SWP achieve the best results for medium and high traffic load.

7.3.2.2 Sprint Topology

For the Sprint network topology, Fig. 7.3 shows that MIRA algorithm performs better under low and medium traffic load. For example, at low traffic load, MIRA achieves an increase of 8.3dB in averaged PSNR when compared to MHA algorithm. In contrast, at high traffic load, the WSP algorithm shows a slight improvement when monitoring the average PSNR as compared to other routing algorithms.

When looking at maximizing the throughput for QoS-based video flows, it has been noticed that MIRA achieves better results than other algorithms under low and medium traffic load. However, as the traffic load increases from low to high, the SWP algorithm maximizes the system throughput for the QoS-based traffic. In terms of minimizing the packet loss for QoS-based flows, MIRA outperforms other algorithms under low and medium traffic load. By increasing the traffic load, SWP algorithm achieves better results. In terms of minimizing the latency for QoS-based flows, MIRA performs the best for low traffic load while SWP achieves the best results for medium and high traffic load.

7.3.2.3 AT&T Topology

Table 7.5 shows in general that the throughput level decreases considerably as the traffic load increases, while the packet loss and latency increases correspondingly. As the network load increases on the same network topology, the links experiences higher congestion rate, increasing at same time, the latency and packet drop rate of the corresponding flows. Fig. 7.3 indicates that as the traffic load increases from low to high, MIRA algorithm achieves a decrease in average PSNR of about 8.8dB. On the other hand, MHA, WSP and SWP algorithms obtain higher decrease of about 10.2dB, 12.7dB, and 13dB, respectively.

The measurement shows that the WSP algorithm achieves the maximum throughput for QoS-based flows under low traffic. When the traffic load becomes medium, MIRA gets the highest throughput level. If the traffic load is high, then MHA achieves better results when compared to other candidates. In terms of minimizing the packet loss for QoS-based flows, SWP achieves the best results under low and high traffic load, while MIRA reaches better results under medium traffic load. In terms of minimizing the latency for QoS-based flows, WSP performs the best for low traffic. For medium traffic, MIRA performs better than other algorithms, while SWP achieves the best results for high traffic load.

7.3.3 Impact on the QoS-based Video Traffic

This section presents the impact of routing algorithms on the PSNR for the QoS-based video traffic. Fig. 7.3 shows the average PSNR for the traffic of quality services. At the low traffic load under GetNet and Sprint networks, it can be seen that MIRA algorithm performs better in comparison to other routing algorithms. For example, in low traffic load under GetNet network, MIRA algorithm achieves an increase of 4.2dB when compared to MHA algorithm. In fact, MIRA algorithm attempts as much as possible to avoid placing the route requests along the links that leads to highly probable congestion. On the other hand, the results shows as well that the WSP and SWP algorithms have a close performance to MIRA algorithm under low traffic load and AT&T network topology. The WSP and SWP algorithms try to balance the network loads to avoid network bottlenecks.

As the traffic load becomes high, the average PSNR decreases and the routing algorithms behave differently in terms of the PSNR performance. For example, the WSP and SWP routing algorithms show similar results under GetNet network. Under the Sprint topology, the WSP algorithm gets better results than other algorithms as it reaches an increase in PSNR performance of about 2.9dB when compared to MHA algorithm. In contrast, MIRA algorithm performs better than other algorithms as it shows an increase of 5.2dB when compared to MHA algorithm.

7.4 Chapter Summary

A vital peripheral within the networking landscape is represented by the routing algorithm as it efficiently routes the flows over the underlying network. There is a wide range of routing algorithms in networking, each with different properties and purpose. The choice of routing algorithm can heavily impact the QoS provisioning within multimedia-based SDNs. In this chapter, a comprehensive performance evaluation is studied for four state-of-the-art centralized routing algorithms (MHA, WSP, SWP and MIRA) over multimedia-based SDN. The experimental setup is demonstrated under a realistic environment with dynamic network conditions and topology. The four algorithms were implemented and evaluated by using an experimental setup based on Mininet, Floodlight controller and Open vSwitch switches. Several scenarios are considered to demonstrate the impact of the state-of-the-art routing algorithms under realistic conditions, on QoS-based video traffic in terms of throughput, packet loss, PSNR, rejection ratio, etc.

This chapter presents the study of the research contribution that leads to the motivation of the proposed reinforcement learning approach for enabling QoS over PBNM-based SDN networks. The aim of this chapter is to assess the performance of four routing algorithms under different dynamic network scenarios like network loads and topology. It is attributed to how different network load and network topology can have an impact on the routing algorithm performance. The results of the study highlights the importance of the integration of learning-based methods and other entire QoS-based solutions under SDN-based environments. The results of this extensive performance evaluation study show that there is no single routing algorithm that would perform best under highly dynamic network conditions and demonstrates the applicability of machine learning in this context. The results draw benefit to the entire thesis by studying and understanding the experimental setup scenario that is used for the next step towards a machine learning-based traffic management solution in the framework. In general, the results show that there is no one single routing algorithm that can perform the best for all considered scenarios and networking conditions. It was noticed that the routing algorithms perform differently under various traffic load and network topology. The following chapter discusses the integration of an intelligent traffic management scheme that can adapt to the changeable networking conditions (traffic load, topology, etc.) and decide the most convenient routing algorithm to be used each time.

Chapter 8

RL-based Decision Making for Routing Algorithms over SDN

This chapter presents the performance evaluation of the research contribution: RL-based Decision Making for Routing Algorithms under Policy-based SDN Environment. *The chapter presents the details of the experimental setup environment, the scenarios and case studies used for the performance evaluation and provides a comprehensive discussion on the results obtained*

8.1 Experimental Setup Environment

The experimental setup is used to evaluate the RL-based method (as described in 4.4). The overall test-bed is comprised of three main elements: (i) Mininet [178] - used to emulate the SDN data plane; (ii) external Floodlight OpenFlow controller [53] - 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 are collected and are stored for post performance comparison. In the work of this chapter, the entire experiment is hosting on a powerful machine to accommodate the traffic load. The SDN controller and the entire application layer run on a virtual computer (2.2GHz multiprocessor of 4 CPU units with memory size of 16GB), while the Mininet test-bench is running on another virtual machine (2.2GHz multiprocessor of 4 CPU units with memory size of 32GB). Each virtual machine is running Linux-Ubuntu Server. Open vSwitch [202] 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.

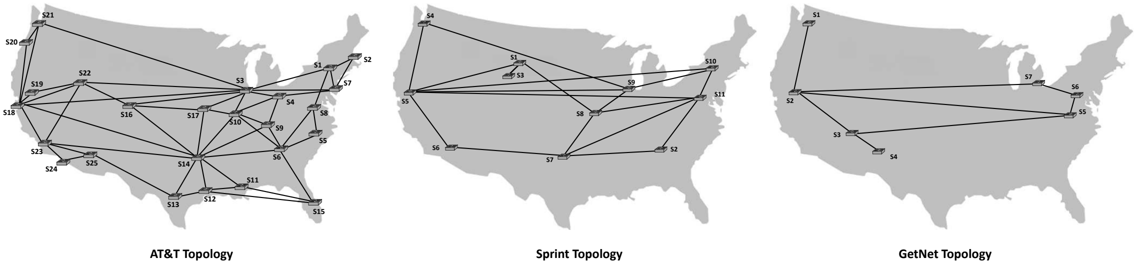

Figure 8.1: Experimental Setup using the following network topologies: AT&T (large-scale topology), Sprint (middle-scale topology), and GetNet (small-scale topology)

The performance evaluation of the proposed method is performed under dynamic network conditions and over three realistic network topologies, modeled by an undirected graph $G(V, E)$, selected from Internet Topology Zoo as illustrated in Fig. 8.1:

- 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.

In the experimental setup, two types of services are generated: the flows of QoS-based multimedia services F_{qos} and the flows of the background services F_{bkg}. Under the two service types, four traffic classes $v \in \{HD\ video,\ SD\ video,\ HTTP,\ FTP\}$ are emulated: live High-Definition (HD) video streaming as part of the QoS-based multimedia services and buffered Standard-Definition (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 [203]. The FFMPEG tool is an open-source library that is used to convert between arbitrary sample rates and re-size the audio and video data separately. On the other hand, HTTP and FTP traffic are generated using Ostinato [200] traffic generator tool. In order to create a realistic environment different traffic mix and load on the networks are considered.

According to Cisco forecast, video traffic volume will reach 82% of all IP traffic by the year 2022 [29]. Based on this 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 facts in [177], the total volume of 82% for video traffic can be divided into 63% live HD video and 19% buffered SD video. In this work 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 8.1.

Table 8.1: Parameters of video traffic

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

The HTTP and FTP are built on a client-server model architecture. However, the two communications have differences. An HTTP session is used to access websites based on a sequence of request-response transactions. A web browser represents a client side that sends a request message to the web server. Upon receiving the request, the server sends the web page data. Web page traffic is usually based on smaller amount of data than the FTP traffic. On the other hand, the FTP traffic is used to transfer files over the network incorporating in general larger data to transfer.

The parameters for the HTTP traffic model [2, 3] used are listed in Table 6.2. The HTTP traffic is modeled as ON/OFF period, where the ON period corresponding to the transmission time and the OFF period corresponding to the packet inter-arrival time. For each traffic request, the source and destination host pairs are selected randomly following a uniform distribution. On the other hand, the parameters for the FTP traffic model [3] used are listed in Table 7.2. With each call, the session is used to transfer a file of a random generated size. Several experimental scenarios are considered to validate the performance of the proposed framework. Here the performance of proposed learning based algorithm is to be assessed within SDN under a realistic environment with dynamic network conditions such as different topology, traffic patterns and traffic load. The performance evaluation is done in terms of throughput, packet loss, delay, flow rejection and PSNR [35].

8.2 Network Emulation Scenarios

To evaluate the routing algorithms under dynamic network conditions, a scenario with a mix of QoS and background flows is considered. The total experiment duration is set to 1500 seconds. The destination node is chosen at random other than the source node within the network. In order to maintain the traffic mix ratio based on the statistics provided by Cisco [29], each link in the topology operates at the speed of 1 Mb/s. A larger link capacity in the topology requires a higher number of HTTP and FTP flows to sustain the traffic ratio. In the performance evaluation, the following parameters are considered in order to drive a dynamic network evaluation:

- Network topology size (γ): Three different size of network topologies are employed in the study $\gamma \in \{scale_{small},\ scale_{medium},\ scale_{large}\}$: AT&T (large-scale topology) denoted by $scale_{large}$, Sprint (middle-scale topology) denoted by $scale_{medium}$, and GetNet (small-scale topology) denoted by $scale_{small}$. The network topologies were taken from Internet zoo topology [197].

- Network load (ψ): In order to evaluate the performance of routing algorithms, three network load are employed $\psi \in \{load_{low},\ load_{medium},\ load_{high}\}$. This is realized by adjusting the number of active flows in the network 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 considered: 0.5 (low load) denoted by $load_{low}$, 0.75 (medium load) denoted by $load_{medium}$, and 1.0 (high load) denoted by $load_{high}$. The network load is calculated as follows:

$$\psi = \frac{\sum_{l \in E} \frac{\sum_{f \in F} d_{l,f} \cdot a_f}{C_l}}{n(E)} \tag{8.1}$$

  where $\sum_{f \in F} d_{l,f} \cdot a_f$ is the load over the link, C_l is the link capacity, and $n(E)$ denotes the number of links in set of links E.

- Service type: Traffic flows of two service types are generated (QoS traffic F_{qos} and background traffic F_{bkg}). Under the two service types, four traffic classes are emulated in this study, namely $v \in \{HD\ video,\ SD\ video,\ HTTP,\ FTP\}$. For the guaranteed traffic, live HD video streaming is employed. While the 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.

In general, the service performance can be measured in terms of defined metrics. In this work the following end-to-end parameters are considered that are relevant to the service performance [204,205]. Throughput and packet loss rate are measured according to the definition in 2.1. While the rejection rate is defined as the ratio of the total number of rejected traffic request over the total number of traffic requests. The measured rejection rate $\tilde{c}_v$ that belongs to the traffic class v can be calculated as follows:

$$\tilde{c}_v = \frac{n^{rej}}{n^{rej} + n^{acc}} \times 100 \tag{8.2}$$

where n^{rej} and n^{acc} are the total number of rejected and accepted incoming traffic requests, respectively.

Two traffic services are generated in this work: QoS traffic and background traffic. Each type of service requires different demand of requirements $Q_f \in \{Q_{qos}, Q_{bkg}\}$ [206]. In general, the background traffics do not make strong guarantee compared with QoS traffic. The network makes its best-effort to deliver the packets to the destination [207]. Therefore requirement based on the traffic service and class can be initially separated. In the experiment, between the ingress-egress pairs, a set of requirement demands for the live HD video traffic need to be satisfied. In general, video is considered sensitive to network degradation. In order to satisfy the human perception, video quality becomes noticeable at packet loss of 0.5% and annoying when greater than 2% [30, 208–212]. Similarly, other literature like [213–215] indicates that video conferencing with a packet loss between 1% and 2.5% is considered acceptable and above 4-6% packet loss video conferencing becomes irritating. Based on this, the maximum acceptable packet loss rate $q_{qos,loss}$ is defined for the live HD video traffic of QoS service to 1%. On the other hand, in order to meet an acceptable video perception the maximum acceptable packet loss of background traffic $q_{bkg,loss}$ is defined for the buffered SD video traffic by 2%. Other background traffic like HTTP and FTP shall have a guarantees of zero packet loss rate [30,209].

The minimum throughput requirement $q_{v,thr}$ is obtained based on the average video bit-rate and the corresponding packet loss requirement. It is calculated based on:

$$q_{v,thr} = (100\% - q_{v,loss}) \times br_v \tag{8.3}$$

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 8.2 illustrates the set of requirements for 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.

Table 8.2: 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%

8.3 Network Emulation Results and Analysis

The proposed RL-based framework as described in section 4.4.4.2 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 [216]. 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 4.4.1, 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.

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 [20]. 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, ϵ-greedy algorithm is used to give a chance to execute random action. In order to apply a fair exploration-exploitation trade-off, the ϵ-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 ϵ-greedy, the ϵ 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. The Q-learning algorithm updates the Q-table based on the equation described in 4.6. 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 [20].

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 $\alpha = 0.01$. 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. 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.

8.3.1 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.

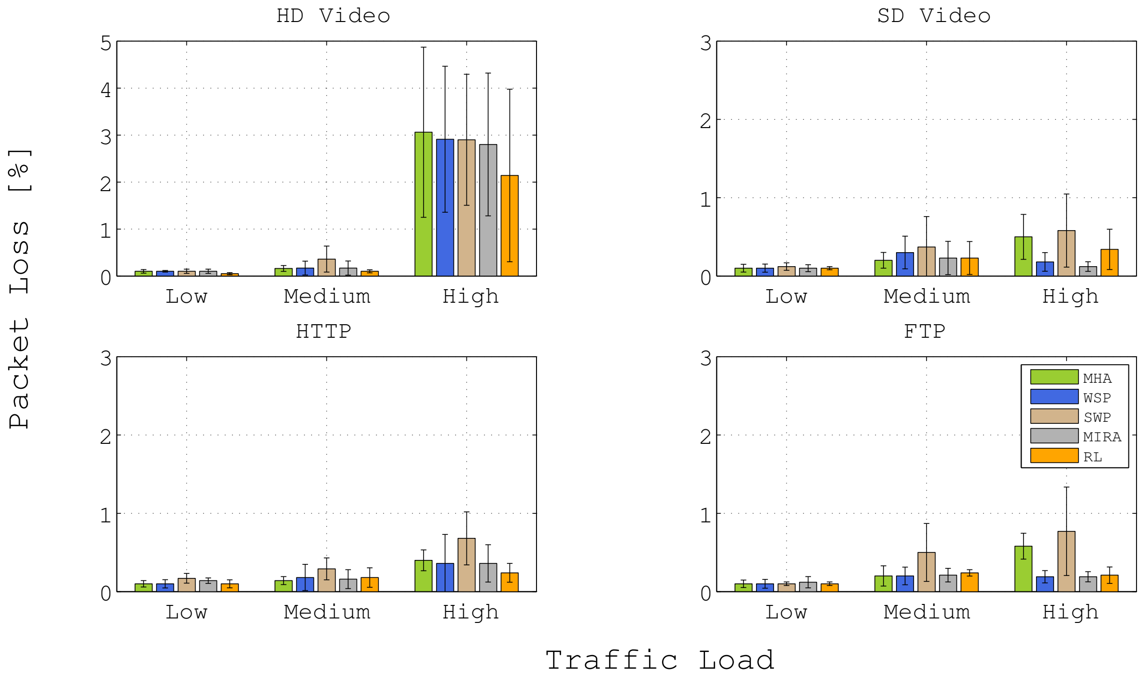

Figure 8.2: GetNet network topology: Packet loss of the traffic classes under different traffic loads

8.3.1.1 GetNet Topology

The results for the small scale network like GetNet topology are illustrated in Tables 8.3, 8.4, and 8.5 and Figures 8.2 and 8.3. The results in Table 8.3 and Figure 8.2 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 8.2. 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 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 8.2). Consequently, it can be seen that under the small scale network, all the solutions maintain an *Excellent* QoE (see Table 8.5) 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*.

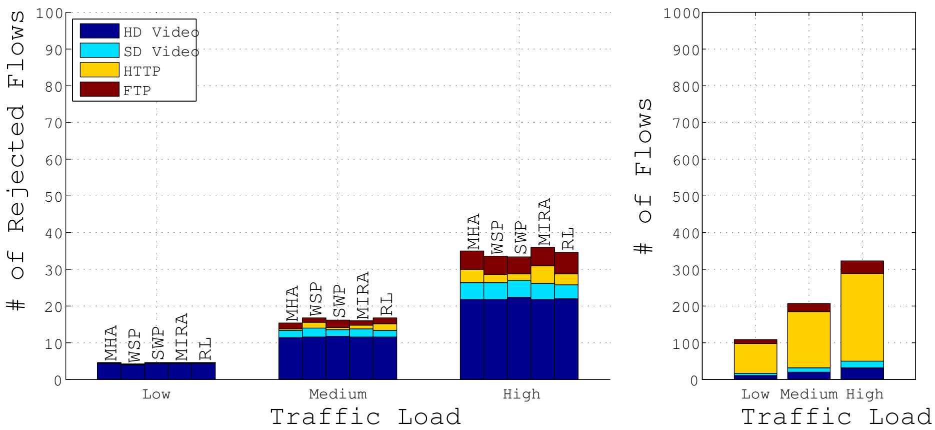

Figure 8.3: GetNet network topology: The total number of rejected flow and the total number of flows that are generated in the experiment test

On the other hand, Figure 8.3 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, 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.

8.3.1.2 Sprint Topology

The results for the medium scale network like Sprint topology are illustrated in Tables 8.3, 8.4, and 8.5 and Figures 8.4 and 8.5. Table 8.3 and Figure 8.4 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

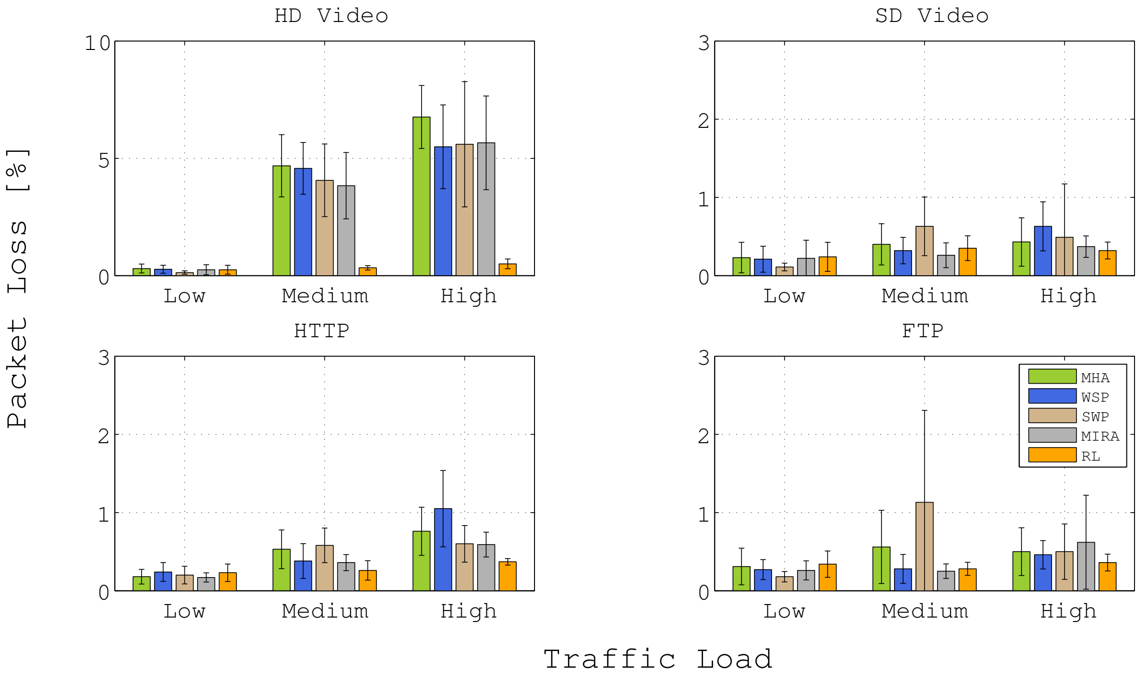

Figure 8.4: Sprint network topology: Packet loss of the traffic classes under different traffic loads

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 8.5). 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 a 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 2.1 and Table 8.5) for the QoS-based traffic class, without penalizing the

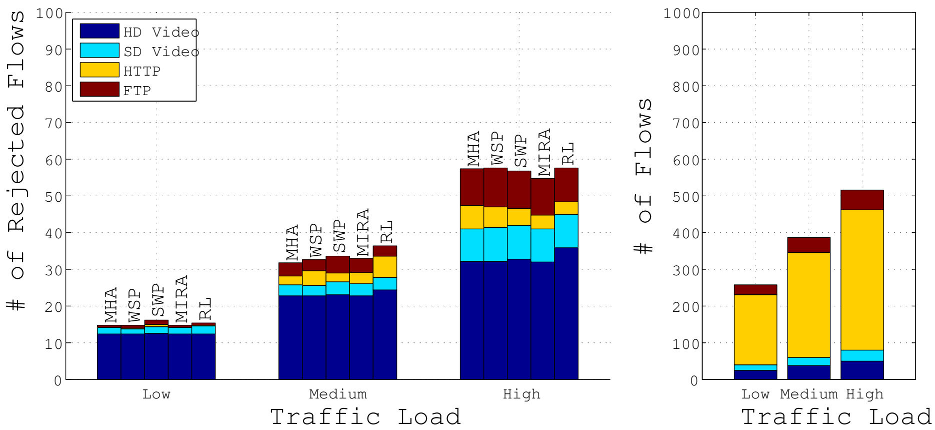

Figure 8.5: Sprint network topology: The total number of rejected flow and the total number of flows that are generated in the experiment test

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.

8.3.1.3 AT&T Topology

The results for the large scale network like AT&T topology are illustrated in Tables 8.3, 8.4, and 8.5 and Figures 8.6 and 8.7. 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 8.3 and Figure 8.6 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 8.5, 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 8.7 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

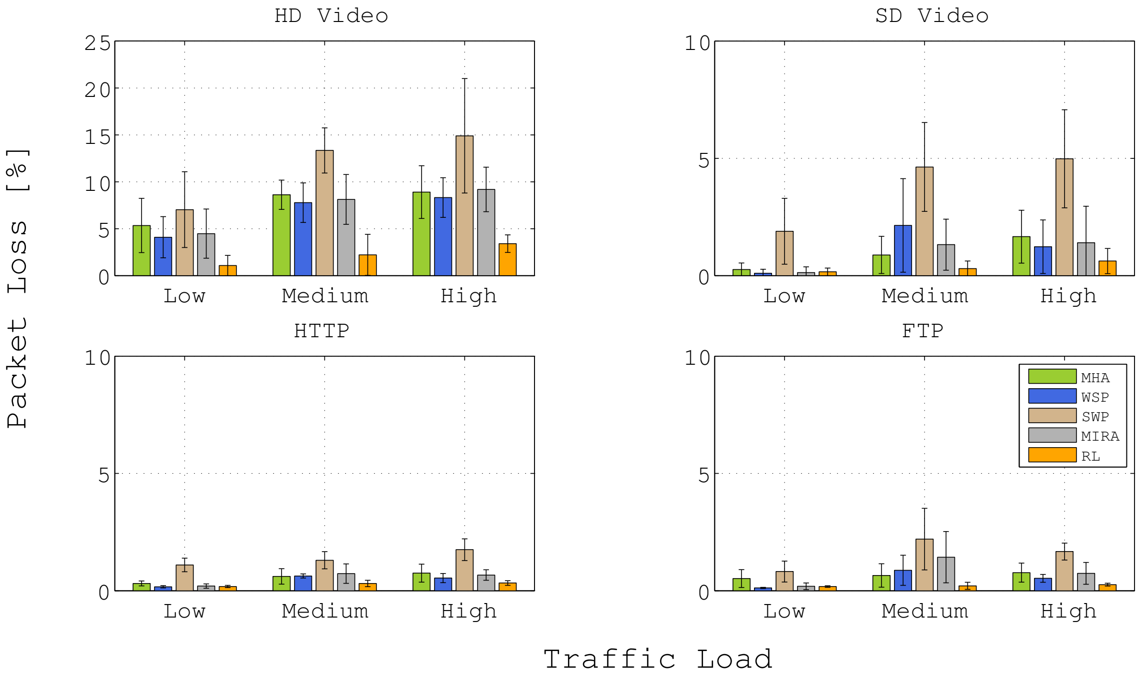

Figure 8.6: AT&T network topology: Packet loss of the traffic classes under different traffic loads

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 2.1) 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.

8.3.2 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.

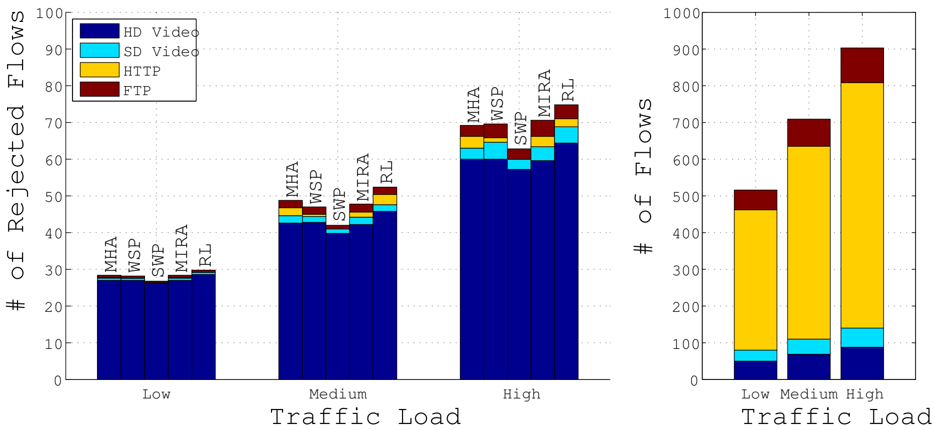

Figure 8.7: AT&T network topology: The total number of rejected flow and the total number of flows that are generated in the experiment test

8.3.2.1 Low Traffic Load

As depicted in Figures 8.2, 8.4, and 8.6, 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 8.5). 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 8.4). The results show that under low traffic load, as the network size increases, the proposed RL-based solution 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 8.3, 8.5, and 8.7 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

8.5, 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 8.3 and the Figures 8.2, 8.5 and 8.7, 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 8.2, 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.

8.3.2.2 Medium Traffic Load

For the medium load, the results in Figures 8.2, 8.4, and 8.6 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 8.3). 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 8.3, 8.5, and 8.7 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 8.5) 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 8.2, the RL-based method conforms to the requirement under the GetNet and Sprint network. Though, under the AT&T network, the results in Figure 8.6 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.

8.3.2.3 High Traffic Load

Figures 8.2, 8.4 and 8.6 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 cannot 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 8.5) 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.

8.3.3 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 8.6. 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.

8.4 Chapter Summary

This chapter presents 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. Several scenarios are considered under realistic network conditions.

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.

Table 8.3: 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 ±3.96	651 ±3.96	624 ±10.2	650 ±4.16	651 ±1.17	626 ±7.95	650 ±3.62	649 ±4.55	625 ±8.79	650 ±4.12	651 ±2.17	626 ±7.82	650 ±4.11	651 ±3.56	631 ±12.57
SD	238 ±1.81	235 ±0.61	232 ±1.57	239 ±1.18	235 ±0.78	233 ±1.71	238 ±1.79	234 ±1.58	232 ±1.95	239 ±1.76	233 ±3.25	233 ±1.32	238 ±1.1	235 ±0.87	233 ±0.45
HTTP	15.8 ±1.5	15.3 ±0.98	16.2 ±0.5	16.4 ±1.5	15.3 ±1.39	16.3 ±0.55	15.9 ±1.73	15.6 ±0.95	16.1 ±0.39	16 ±1.73	15.7 ±3.25	16.3 ±0.78	15.8 ±1.79	15.5 ±1.24	15.9 ±0.6
FTP	172.9 ±3.95	185.6 ±2	178.4 ±3.46	173.6 ±4.55	184.4 ±1.85	177.3 ±4.6	172.2 ±3.67	184.4 ±2.49	178.8 ±3.15	173 ±4.15	184 ±2.04	179.1 ±5.25	172 ±3.6	184.4 ±3.24	178.6 ±4.55
Sprint															
HD	648 ±1.17	620 ±11.0	596 ±7.4	648 ±1.18	620 ±10.1	606 ±8.6	649 ±0.76	624 ±11.6	606 ±1.8	648 ±1.52	626 ±11.9	606 ±13.4	646 ±1.52	643 ±2.0	638 ±5.4
SD	236 ±0.51	235 ±1.58	233 ±1.3	236 ±0.53	235 ±0.93	233 ±2.2	236 ±0.16	234 ±1.3	234 ±1.5	236 ±0.6	235 ±1.3	234 ±2.5	236 ±0.51	234 ±1.15	235 ±1.77
HTTP	15.2 ±1.6	16 ±1.0	14.6 ±1.5	15.2 ±1.25	15.7 ±0.84	14.4 ±1.9	15.3 ±1.27	16.1 ±0.84	14.6 ±1.85	15.2 ±1.42	15.7 ±1.0	14.5 ±2.0	15.3 ±1.37	15.9 ±1.0	14.3 ±1.7
FTP	186.7 ±3.5	177.8 ±7.6	161.7 ±4.9	185.7 ±2.6	177.8 ±5.9	165.7 ±5.0	186.8 ±3.2	174.8 ±5.2	164 ±3.1	186.6 ±2.72	178.7 ±7.3	166.3 ±4.7	186.3 ±2.5	177.8 ±7.6	167.9 ±2.7
AT&T															
HD	616 ±18.9	594 ±9.9	605 ±18.9	624 ±14.9	604 ±16.1	607 ±14.6	617 ±28.4	576 ±20.3	567 ±42.4	622 ±17.5	604 ±20.8	601 ±16.2	653 ±10.0	630 ±8.4	633 ±8.7
SD	232 ±0.66	233 ±3.2	236 ±2.8	232 ±0.87	230 ±4.3	237 ±2.8	231 ±3.5	227 ±4.72	232 ±4.87	232 ±0.88	232 ±2.8	236 ±4.0	232 ±0.7	236 ±0.42	239 ±1.1
HTTP	11.3 ±0.5	11.5 ±2.1	13.3 ±3.8	10.8 ±0.35	11.6 ±2.6	12.4 ±3.0	11.1 ±0.23	12.2 ±2.7	13.9 ±4.0	11.3 ±0.38	11.4 ±1.6	12.6 ±3.5	11.3 ±0.3	10.8 ±0.54	13.1 ±4.1
FTP	162.5 ±2.9	155.7 ±2.1	149 ±1.0	164.2 ±3.2	155 ±3.3	149.4 ±1.85	160.9 ±3.8	149.7 ±3.5	144.6 ±1.6	162 ±5.3	154 ±2.1	148 ±1.8	164 ±3.72	155.9 ±3.3	149.7 ±0.866

Table 8.4: 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

Table 8.5: 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

Table 8.6: 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

Chapter 9

Conclusions and Future Work

This chapter presents a summary of the work and the key contributions of this research. It is followed by a discussion about the future work and it highlights the suggestions for deeper analysis and improvement under the taken research.

9.1 Summary of Contributions

The research proposes a framework for enabling end-to-end QoS provisioning over SDN-based environments. Nowadays, QoS provisioning represents a vital entity in the network infrastructure in order to fulfill the business needs of the Internet Service Providers. Moreover, the emergence of SDN-based networks opens up new opportunities for new features in terms of management and programming compared to the traditional networks.

In this regard, the work presented in this thesis brings the following main contributions:

- **Measurement Collection and Probabilistic-based QoS Routing:** The proposed solution described in 4.2.1.3 proposes the use of a compression technique for SDN-based networks to reduce the control plane overhead. The contribution of this work is introduced by the proposed architecture which facilitates the delivery of QoS metric data with less overhead and high accuracy. This is achieved by the use of the sparse techniques. The performance evaluation results presented in Chapter 5 show that by employing a CR = 0.333 the controller recovers the sparse data with an SNR of 36.269dB and an NMAE of 0.01063. Moreover, in terms of communication overhead cost the results show that the proposed compression-based technique reduces the overhead significantly.

  On the other hand, the solution given in 4.2.2 proposes, BaProbSDN a probabilistic-based QoS routing algorithm for SDNs. BaProbSDN makes use of a Bayes' theorem to determine the link probability in terms of bandwidth availability. In order to decrease the overhead on the control plane between the SDN switches and the controller, BaProbSDN makes use of a threshold-based triggering link update policy combined with a hold-down timer. The performance evaluation results presented in Chapter 5 show that the overhead can be greatly reduced with less significant impact on the performance in terms of bandwidth blocking rate. Moreover the proposed BaProbSDN algorithm was compared against the WSP algorithm in the presence of link state update policy. The results show that BaProbSDN can achieve up to 7.41% decrease in the bandwidth blocking rate when compared to WSP, for a threshold value of 0.5, HDT=10s, and a reduction of 95.38% in the control messages overhead.

- **Policy-based Network Management Framework:** The solution given in 4.3 proposes a policy-based network management framework over SDN for QoS provisioning. The proposed solution considers the route management for finding the optimal route for QoS flows and for rerouting and rate limiting the disturbing flows in case of policy violation. The PBNM framework was designed and implemented over the SDN architecture and the OpenFlow protocol is used to collect information and configure the underlying SDN network switches. The performance evaluation results presented in Chapter 6 show that by applying QoS policies for bandwidth and loss rate, the PBNM framework can dynamically reconfigure the network state by rerouting and rate limiting the best-effort background flows to ensure QoS provisioning for the priority flows. The experimental results show that the proposed PBNM-based SDN framework outperforms the default SDN in terms of throughput, packet loss rate and latency. Moreover, both proposed approaches, like rerouting and rate limiting demonstrate comparable performance. The results show that the proposed PBNM-based SDN framework for rerouting can achieve up to 94% increase in the average PSNR when compared to the default SDN, increasing the user perceived quality from *Poor* to *Excellent*.

- **RL-based Decision Making for Routing Algorithm:** The comprehensive study presented in Chapter 7 evaluates the performance of four state-of-the-art routing algorithms MHA, WSP, SWP, and MIRA over realistic multimedia-based SDN environments with dynamic network conditions and topology. The results indicate that there is no one single routing algorithm that can perform best under all considered scenarios and networking conditions. These findings led to the solution described in 4.4 that proposes a machine RL-based method by making use of RL under SDN-based networks. As SDN comes with key features such as the centralization and programmability via standardized interface, this makes it possible to implement the proposed RL-based solution in a centralized way under the SDN network architecture. The proposed framework contains a continuous monitoring scheme to build a global centralized view of the network state. It collects periodically the flow statistics from the network switches. The system is first trained while it interacts with the surrounding environment from the experiments of past trails. Based on the trained Q-table, the system exploits the most suitable routing algorithm to be applied on the underlying network. The proposed RL-based method enables dynamic routing decisions and it reroutes the QoS-based traffic flows in order to find the best solution to the problem by choosing the right routing algorithm to be applied from a set of routing algorithms, i.e., MHA, WSP, SWP, and MIRA.

The performance evaluation results are presented in Chapter 8 where the RL-based method was evaluated over realistic SDN-based network environment with dynamic network conditions and topology and compared against other centralized routing algorithms (MHA, WSP, SWP and MIRA). The results show that the proposed RL-based solution outperforms the other state-of-the-art routing algorithms 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 high 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.

9.2 Future Work

Future work is a place for further suggestions and ideas to analyze deeper the proposed solution with particular techniques and experiment tools. The topics addressed in this thesis (as described in Chapter 4) offer different directions to research in the future. In the current research the PBNM architecture is mapped into the entire framework in its three layers: PIP, PDP, and PEP layer. In the context of PBNM-based SDN framework, future work could extend the functionality of the proposed framework by making use of policy language and refinement. This shall add more abstraction layer and automation principle to the framework. By this, the network administrator shall deal with a higher abstracted language with less detail about the networking infrastructure. Due to the lack of physical network resources and big scaled real network infrastructure, the proposed framework has been evaluated entirely under simulation environment with the help of various networking simulation tools (such as Mininet, Ostinato and Matlab) while keeping realistic conditions as much as possible. An essential step in the further assessment is the evaluation with real data under a real SDN network.

Regarding the entire applications, an idea for further enhancement in terms of faster algorithm is to employ parallelism concept in the implementation of the intelligence application core. On the other hand, due to many factors: the lack of time, the availability of necessary features and the dependencies and mismatch between modules, the current network is realized with OpenFlow 1.3 protocol. A higher version of OpenFlow protocol can be employed and realized in the current framework. As a proof of concept, the routing algorithms in this research consider the bandwidth availability metric to find the feasible path. From a theoretical point of view, the centralized routing algorithms can be extended by considering the delay constraint in the routing calculation. Though, even if it was not addressed here due to the computation limitation, this represents an interesting challenge for future work. With respect to the monitoring scheme under SDN network, the continuous monitoring (as described in 4.2.1.1) is applied currently in the framework in order to build a global image of the underlining networking devices in terms of the flow throughput and packet loss. An approach which is not addressed in the thesis under the dynamic simulation experimental setup is the monitoring using the link state update in 4.2.1.2 and compression based technique in 4.2.1.3. Due to the reason given earlier in section 5.3, the test was limited under simulation tool based on Matlab. An interesting study is its application and interaction with other component in the framework such as the RL-based solution.

A further topic that can be addressed potentially for future work is the use of multiple SDN controllers, which gives the chance to scale the concept on wider network environment. Though the synchronization by involving intra- and inter-domain communications between multiple SDN controllers becomes challenging, while maintaining the research goals. In SDN-enabled network, the multi-controller architecture can improve the performance in terms of reliability, scalability, and availability when compared to the single controller. However this increases more network complexity and opens new challenges such as the controller placement problem. To this end, The proposed solution in this research can be scaled with multi-controller architecture, to achieve this it is essential to share and synchronize globally the entire training data and network view between the controller, and the same operational logic are duplicated between the controllers. By this, many challenges are raised like the propagation latency and the load balancing are while placing the controllers in the network.

Bibliography

[1] Thomas Zinner, Osama Abboud, Oliver Hohlfeld, Tobias Hossfeld, and Phuoc 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.

[2] Hyoung-Kee Choi and John O Limb. A behavioral model of web traffic. In *Network Protocols, 1999.(ICNP'99) Proceedings. Seventh International Conference on*, pages 327–334. IEEE, 1999.

[3] Vladimir Deart, Vladimir Mankov, and Alexander Pilugin. HTTP Traffic Measurements on Access Networks, Analysis of Results and Simulation. In *Smart Spaces and Next Generation Wired/Wireless Networking*, pages 180–190. Springer, 2009.

[4] Jeff Zhuang, Louay Jalloul, Robert Novak, and Jeongho Park. IEEE 802.16 m evaluation methodology document (EMD). *IEEE802*, 16, 2008.

[5] Hyojoon Kim and Nick Feamster. Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2):114–119, 2013.

[6] Adrian Lara, Anisha Kolasani, and Byrav Ramamurthy. Simplifying network management using software defined networking and openflow. In *Advanced Networks and Telecommuncations Systems (ANTS), 2012 IEEE International Conference on*, pages 24–29. IEEE, 2012.

[7] Hewlett-Packard. Primer on Policy Based Network Management, 1999.

[8] Nick McKeown. Software-defined networking. *INFOCOM keynote talk*, 17(2):30–32, 2009.

[9] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[10] ITU-T. *ITU-T Recommendation E.800 (08/94): Terms and definitions related to quality of service and network performance including dependability*, 1994.

[11] Global Mobile Data Traffic Forecast. Cisco visual networking index: global mobile data traffic forecast update, 2017–2022. *Update*, 2017:2022, 2019.

[12] Andrea Westerinen, John Schnizlein, John Strassner, Mark Scherling, Bob Quinn, Shai Herzog, A Huynh, Mark Carlson, Jay Perry, and Steve Waldbusser. Terminology for policy-based management. Technical report, Internet Engineering Task Force, 2001.

[13] R Yavatkar, D Pendarakis, and R Guerin. Rfc 2753–a framework for policy-based admission control, ieft, internet engineering task force std., january 2000.

[14] Kevin Carey, Kevin Feeney, and Dave Lewis. State of the art: Policy techniques for adaptive management of smart spaces. *State of the Art Surveys*, pages 58–66, 2003.

[15] Ian W Marshall, Hamid Gharib, Jim Hardwicke, and Chris Roadknight. A novel architecture for active service management. In *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*, pages 795–810. IEEE, 2001.

[16] Marcial Porto Fernandez, Ade CR Pedroza, and José Ferreira de Rezende. Qos provisioning across a diffserv domain using policy-based management. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 4, pages 2220–2224. IEEE, 2001.

[17] Leonidas Lymberopoulos, Emil Lupu, and Morris Sloman. An adaptive policy based management framework for differentiated services networks. In *Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on*, pages 147–158. IEEE, 2002.

[18] John Strassner. *Policy-based network management: solutions for the next generation.* Morgan Kaufmann, 2003.

[19] Luiz A DaSilva. Pricing for qos-enabled networks: A survey. *IEEE Communications Surveys & Tutorials*, 3(2):2–8, 2000.

[20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press Cambridge, 1998.

[21] Ahmed Al-Jawad, Purav Shah, Orhan Gemikonakli, and Ramona Trestian. Compression-based technique for sdn using sparse-representation dictionary. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pages 754–758. IEEE, 2016.

[22] Ahmed Al-Jawad, Ramona Trestian, Purav Shah, and Orhan Gemikonakli. Baprobsdn: A probabilistic-based qos routing mechanism for software defined networks. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5. IEEE, 2015.

[23] Ahmed Al-Jawad, Purav Shah, Orhan Gemikonakli, and Ramona Trestian. Policy-based qos management framework for software-defined networks. In *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2018.

[24] Ahmed Al-Jawad, Ioan-Sorin Comsa, Purav Shah, Orhan Gemikonakli, and Ramona Trestian. An innovative reinforcement learning-based framework for quality of service provisioning over multimedia-based sdn environments. *IEEE Transactions on Broadcasting*, 2021.

[25] Ahmed Al-Jawad, Purav Shah, Orhan Gemikonakli, Ioan-Sorin Comşa, and Ramona Trestian. Performance evaluation of routing strategies over multimedia-based sdns under realistic environments. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pages 160–164. IEEE, 2020.

[26] Ahmed Al-Jawad, Purav Shah, Orhan Gemikonakli, and Ramona Trestian. Learnqos: A learning approach for optimizing qos over multimedia-based sdns. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6. IEEE, 2018.

[27] Ying Loong Lee, Jonathan Loo, and Teong Chee Chuah. Modeling and performance evaluation of resource allocation for lte femtocell networks. In *Modeling and Simulation of Computer Networks and Systems*, pages 683–716. Elsevier, 2015.

[28] Zheng Wang and Jon Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14:1228–1234, 1996.

[29] VNI Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1:1, 2018.

[30] Rafael Asorey Cacheda, Daniel Castro García, Antonio Cuevas, Francisco Javier González Castano, Javier Herrero Sánchez, Georgios Koltsidas, Vincenzo Mancuso, José Ignacio Moreno Novella, Seounghoon Oh, and Antonio Pantò. Qos requirements for multimedia services. In *Resource management in satellite networks*, pages 67–94. Springer, 2007.

[31] Stefan Winkler and Praveen Mohandas. The evolution of video quality measurement: From psnr to hybrid metrics. *IEEE transactions on Broadcasting*, 54(3):660–668, 2008.

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

[33] Stefan Winkler. Perceptual video quality metrics—a review. In *Digital video image quality and perceptual coding*, pages 155–180. CRC press, 2017.

[34] Thiow Keng Tan, Rajitha Weerakkody, Marta Mrak, Naeem Ramzan, Vittorio Baroncini, Jens-Rainer Ohm, and Gary J Sullivan. Video quality evaluation methodology and verification testing of hevc compression performance. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1):76–90, 2015.

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[36] Wolfgang Braun and Michael Menth. Software-defined networking using openflow: Protocols, applications and architectural design choices. *Future Internet*, 6(2):302–336, 2014.

[37] Manar Jammal, Taranpreet Singh, Abdallah Shami, Rasool Asal, and Yiming Li. Software defined networking: State of the art and research challenges. *Computer Networks*, 72:74–98, 2014.

[38] OpenFlow. *Openflow switch specification, version 1.3.0*, 2012.

[39] Anton Beloglazov and Rajkumar Buyya. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing*, pages 826–831. IEEE Computer Society, 2010.

[40] Anshul Gandhi, Yuan Chen, Daniel Gmach, Martin Arlitt, and Manish Marwah. Minimizing data center SLA violations and power consumption via hybrid resource provisioning. In *Green Computing Conference and Workshops (IGCC), 2011 International*, pages 1–8. IEEE, 2011.

[41] R Yavatkar, D Pendarakis, and R Guerin. IETF RFC 2753: A framework for policy based admission control, 2000.

[42] Murat Karakus and Arjan Durresi. Quality of service (qos) in software defined networking (sdn): A survey. *Journal of Network and Computer Applications*, 80:200–218, 2017.

[43] Atanas Mirchev. Survey of concepts for qos improvements via sdn. *Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM)*, 33:1, 2015.

[44] Jeffrey M Jaffe. Algorithms for finding paths with multiple constraints. *Networks*, 14(1):95–116, 1984.

[45] Qingming Ma and Peter Steenkiste. Quality-of-service routing for traffic with performance guarantees. In *Building QoS into Distributed Systems*, pages 115–126. Springer, 1997.

[46] Murali Kodialam and TV 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)*, volume 2, pages 884–893. IEEE, 2000.

[47] Koushik Kar, Murali Kodialam, and T VIEEE Lakshman. Minimum interference routing of bandwidth guaranteed tunnels with mpls traffic engineering applications. *IEEE Journal on selected areas in communications*, 18(12):2566–2579, 2000.

[48] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[49] ONF: Open Networking Foundation. `https://www.opennetworking.org/`. Accessed on Aug 1, 2020.

[50] Open Networking Fundation. Software-defined networking: The new norm for networks. *ONF White Paper*, 2(2-6):11, 2012.

[51] Faqir Zarrar Yousaf, Michael Bredel, Sibylle Schaller, and Fabian Schneider. NFV and SDN—Key technology enablers for 5G networks. *IEEE Journal on Selected Areas in Communications*, 35(11):2468–2478, 2017.

[52] Zainab Zaidi, Vasilis Friderikos, Zarrar Yousaf, Simon Fletcher, Mischa Dohler, and Hamid Aghvami. Will sdn be part of 5g? *IEEE Communications Surveys & Tutorials*, 20(4):3220–3258, 2018.

[53] Floodlight openflow controller. `http://www.projectfloodlight.org/`. Accessed on Aug 1, 2020.

[54] Maciej Kuźniar, Peter Perešíni, and Dejan Kostić. What you need to know about sdn flow tables. In *International Conference on Passive and Active Network Measurement*, pages 347–359. Springer, 2015.

[55] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, 16(3):1617–1634, 2014.

[56] Frank Dagenhardt, Jose Moreno, and Bill Dufresne. *Deploying ACI: The complete guide to planning, configuring, and managing Application Centric Infrastructure*. Cisco Press, 2018.

[57] Cisco Application Centric Infrastructure. `https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-736899.html?dtid=osscdc000283`. Accessed on Aug 1, 2020.

[58] Masayoshi Kobayashi, Srini Seetharaman, Guru Parulkar, Guido Appenzeller, Joseph Little, Johan Van Reijendam, Paul Weissmann, and Nick McKeown. Maturing of openflow and software-defined networking through deployments. *Computer Networks*, 61:151–175, 2014.

[59] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. B4: Experience with a globally-deployed software defined wan. *ACM SIGCOMM Computer Communication Review*, 43(4):3–14, 2013.

[60] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2014.

[61] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014.

[62] Samphel Norden. Inter-domain routing: Algorithms for qos guarantees. *Computer Networks*, 49(4):593–619, 2005.

[63] Chotipat Pornavalai, Goutam Chakraborty, and Norio Shiratori. Qos based routing algorithm in integrated services packet networks. *Journal of High Speed Networks*, 7(2):99–112, 1998.

[64] Giuseppe Araniti, John Cosmas, Antonio Iera, Antonella Molinaro, R Morabito, and Antonino Orsino. Openflow over wireless networks: Performance analysis. In *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pages 1–5. IEEE, 2014.

[65] Slavica Tomovic, Igor Radusinovic, and Neeli 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*, pages 1–6. IEEE, 2015.

[66] Ming-Chieh Lee and Jang-Ping Sheu. An efficient routing algorithm based on segment routing in software-defined networking. *Computer Networks*, 103:44–55, 2016.

[67] Sarah Abdallah, Ayman Kayssi, Imad H Elhajj, and Ali Chehab. Network convergence in sdn versus ospf networks. In *2018 Fifth International Conference on Software Defined Systems (SDS)*, pages 130–137. IEEE, 2018.

[68] Hilmi E Egilmez, S Tahsin Dane, Burak Gorkemli, and A Murat 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*, pages 22–27, 2012.

[69] Hilmi E Egilmez, Seyhan Civanlar, and A Murat Tekalp. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks. *IEEE Transactions on Multimedia*, 15(3):710–715, 2013.

[70] Hilmi E Egilmez, Burak Gorkemli, A Murat Tekalp, and Seyhan Civanlar. Scalable video streaming over openflow networks: An optimization framework for qos routing. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2241–2244. IEEE, 2011.

[71] Tsung-Feng Yu, Kuochen Wang, and Yi-Huai Hsu. Adaptive routing for video streaming with QoS support over SDN networks. In *Information Networking (ICOIN), 2015 International Conference on*, pages 318–323. IEEE, 2015.

[72] Francesco Ongaro, Eduardo Cerqueira, Luca Foschini, Antonio Corradi, and Mario 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*, pages 505–509. IEEE, 2015.

[73] Xianshu Jin, Hwiyun Ju, Sungchol Cho, Boyeong Mun, Cheongbin Kim, and Sunyoung Han. Qos routing design for adaptive streaming in software defined network. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2016 International Symposium on*, pages 1–6. IEEE, 2016.

[74] Yujeong Oh, Jonghun Kim, and Jaiyong Lee. Qos constrained path optimization algorithm in nfv/sdn environment. *ICN 2016*, page 142, 2016.

[75] Siamak Layeghy. Scor: Software-defined constrained optimal routing platform for sdn. *PhD Thesis, School of Information Technology and Electrical Engineering, The University of Queensland*, 2018.

[76] 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)*, pages 1–4. IEEE, 2018.

[77] Haitham Ghalwash and Chun-Hsi Huang. A qos framework for sdn-based networks. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 98–105. IEEE, 2018.

[78] Sharleen Joy Y Go, Cedric Angelo M Festin, and Wilson M Tan. An sdn-based framework for improving the performance of underprovisioned ip video surveillance networks. *Journal of Network and Computer Applications*, 132:49–74, 2019.

[79] Sheng Zhu, Zhen Sun, Yong Lu, Lianming Zhang, Yehua Wei, and Geyong Min. Centralized qos routing using network calculus for sdn-based streaming media networks. *IEEE Access*, 7:146566–146576, 2019.

[80] Shuangyin Ren, Wenhua Dou, and Yu Wang. A deterministic network calculus enabled qos routing on software defined network. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pages 182–186. IEEE, 2017.

[81] Piyawit Tantisarkhornkhet and Warodom Werapun. Qlb: Qos routing algorithm for software-defined networking. In *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 1–6. IEEE, 2016.

[82] Davide Adami, Gianni Antichi, Rosario G Garroppo, Stefano Giordano, and Andrew W Moore. Towards an sdn network control application for differentiated traffic routing. In *2015 IEEE International Conference on Communications (ICC)*, pages 5827–5832. IEEE, 2015.

[83] Steven Van den Berghe, IMEC Giorgio Ventre, and Dimitrios Giannakopoulos. Advanced qos provisioning in ip networks: The european premium ip projects. *IEEE Communications Magazine*, page 3, 2003.

[84] Fuliang Li, Jiannong Cao, Xingwei Wang, and Yinchu Sun. A qos guaranteed technique for cloud applications based on software defined networking. *IEEE access*, 5:21229–21241, 2017.

[85] Hongyan Cui, Chenhang Ma, Wei Lai, Lili Zheng, and Yunjie Liu. Accurate network resource allocation in sdn according to traffic demand. In *4th Int. Conf on Mechatronics, Chemistry and ComputerEngineering (ICMMCCE 2015)*, pages 1166–1175, 2015.

[86] Chenhui Xu, Bing Chen, and Hongyan Qian. Quality of service guaranteed resource management dynamically in software defined network. *Journal of Communications*, 10(11):843–850, 2015.

[87] Harold Owens and Arjan Durresi. Video over software-defined networking (vsdn). In *Network-Based Information Systems (NBiS), 2013 16th International Conference on*, pages 44–51. IEEE, 2013.

[88] Harold Owens II and Arjan Durresi. Video over software-defined networking (vsdn). *Computer Networks*, 92:341–356, 2015.

[89] Jochen W Guck and Wolfgang Kellerer. Achieving end-to-end real-time quality of service with software defined networking. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 70–76. IEEE, 2014.

[90] Jochen W Guck, Amaury Van Bemten, and Wolfgang Kellerer. Detserv: Network models for real-time qos provisioning in sdn-based industrial environments. *IEEE Transactions on Network and Service Management*, 14(4):1003–1017, 2017.

[91] Tsung-Nan Lin, Yang-Ming Hsu, Sheng-Yi Kao, and Po-Wen Chi. Opene2eqos: Meter-based method for end-to-end qos of multimedia services over sdn. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*, pages 1–6. IEEE, 2016.

[92] Muzzamil Aziz, H Amirreza Fazely, Giada Landi, Domenico Gallico, Kostas Christodoulopoulos, and Philipp Wieder. Sdn-enabled application-aware networking for data center networks. In *Electronics, Circuits and Systems (ICECS), 2016 IEEE International Conference on*, pages 372–375. IEEE, 2016.

[93] M Said Seddiki, Muhammad Shahbaz, Sean Donovan, Sarthak Grover, Miseon Park, Nick Feamster, and Ye-Qiong Song. FlowQoS: QoS for the rest of us. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 207–208. ACM, 2014.

[94] Sachin Sharma, Dimitri Staessens, Didier Colle, David Palma, Joao Goncalves, Ricardo Figueiredo, Donal Morris, Mario Pickavet, and Piet Demeester. Implementing quality of service for the software defined networking enabled future internet. In *Software Defined Networks (EWSDN), 2014 Third European Workshop on*, pages 49–54. IEEE, 2014.

[95] Francesco Lucrezia, Guido Marchetto, Fulvio Risso, Michele Santuari, and Matteo Gerola. A proposal for end-to-end qos provisioning in software-defined networks. *International Journal of Electrical and Computer Engineering (IJECE)*, 7(4):2261–2277, 2017.

[96] Alexandre T Oliveira, Bruno Josó CA Martins, Marcelo F Moreno, Alex B Vieira, Antônio Tadeu A Gomes, and Artur Ziviani. Sdn-based architecture for providing qos to high performance distributed applications. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00602–00607. IEEE, 2018.

[97] Alexandre T Oliveira, Bruno José CA Martins, Marcelo F Moreno, Antônio Tadeu A Gomes, Artur Ziviani, and Alex Borges Vieira. Sdn-based architecture for providing quality of service to high-performance distributed applications. *International Journal of Network Management*, page e2078, 2019.

[98] Joaquin Chung, Eun-Sung Jung, Rajkumar Kettimuthu, Nageswara SV Rao, Ian T Foster, Russ Clark, and Henry Owen. Advance reservation access control using software-defined networking and tokens. *Future Generation Computer Systems*, 79:225–234, 2018.

[99] Joaquin Chung, Rajkumar Kettimuthu, Nam Pho, Russ Clark, and Henry Owen. Orchestrating intercontinental advance reservations with software-defined exchanges. *Future Generation Computer Systems*, 95:534–547, 2019.

[100] Alaitz Mendiola, Jasone Astorga, Eduardo Jacob, and Juanjo Unzilla. An advance reservation system to improve network resource utilization in software-defined networks. *IEEE Access*, 8:40512–40535, 2020.

[101] Iris Bueno, José Ignacio Aznar, Eduard Escalona, Jordi Ferrer, and Joan Antoni Garcia-Espin. An opennaas based sdn framework for dynamic qos control. In *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, pages 1–7. IEEE, 2013.

[102] Huu Thanh Nguyen, Anh Vu Vu, Duc Lam Nguyen, Manh Nam Tran, Quynh Thu Ngo, Thu-Huong Truong, Tai Hung Nguyen, Thomas Magedanz, et al. A generalized resource allocation framework in support of multi-layer virtual network embedding based on sdn. *Computer Networks*, 92:251–269, 2015.

[103] Christopher Williams, Elisa Bertino, S Calo D Verma, K Leung, and C Dearlove. Towards an architecture for policy-based management of software defined coalitions. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6. IEEE, 2017.

[104] Md Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba. PolicyCop: an autonomic QoS policy enforcement framework for software defined networks. *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, pages 1–7, 2013.

[105] Shihabur Rahman Chowdhury, Md Faizul Bari, Reaz Ahmed, and Raouf Boutaba. PayLess: A Low Cost Netowrk Monitoring Framework for Software Defined Networks. *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2014.

[106] Cristian Hernandez Benet, Kyoomars Alizadeh Noghani, Andreas Kassler, Ognjen Dobrijevic, and Patrick Jestin. Policy-based routing and load balancing for evpn-based data center interconnections. In *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–7. IEEE, 2017.

[107] Wendong Wang, Ye Tian, Xiangyang Gong, Qinglei Qi, and Yannan Hu. Software defined autonomic QoS model for future Internet. *Journal of Systems and Software*, 110:122–135, 2015.

[108] Wendong Wang, Qinglei Qi, Xiangyang Gong, Yannan Hu, and Xirong Que. Autonomic QoS management mechanism in software defined network. *China Communications*, 11(7):13–23, 2014.

[109] Felipe Volpato, Madalena Pereira Da Silva, Alexandre Leopoldo Gonçalves, and Mario Antonio Ribeiro Dantas. An autonomic qos management architecture for software-defined networking environments. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 418–423. IEEE, 2017.

[110] Georgios Xilouris, Stavros Kolometsos, Carlos Parada, José Bonnet, Panagiotis Stavrianos, Evgenia Kapassa, Marios Touloupou, Dimosthenis Kyriazis, Panagiotis Gouvas, Eleni Fotopoulou, et al. Towards autonomic policy-based network service deployment with sla and monitoring. In *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–2. IEEE, 2018.

[111] Daniel Rosendo, Patricia Takako Endo, Djamel Sadok, and Judith Kelner. An autonomic and policy-based authorization framework for openflow networks. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE, 2017.

[112] QianQian Li, Reza Mohammadi, Mauro Conti, ChuanHuang Li, and XiaoLin Li. Golfengine: Network management system for software defined networking. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 239–246. IEEE, 2017.

[113] Bo Yan, Yang Xu, and H Jonathan Chao. Bigmac: Reactive network-wide policy caching for sdn policy enforcement. *IEEE Journal on Selected Areas in Communications*, 36(12):2675–2687, 2018.

[114] Cataldo Basile, Fulvio Valenza, Antonio Lioy, Diego R Lopez, and Antonio Pastor Perales. Adding support for automatic enforcement of security policies in nfv networks. *IEEE/ACM Transactions on Networking*, 27(2):707–720, 2019.

[115] Adrian Lara and Byrav Ramamurthy. Opensec: Policy-based security using software-defined networking. *IEEE transactions on network and service management*, 13(1):30–42, 2016.

[116] Andreas Voellmy, Hyojoon Kim, and Nick Feamster. Procera: a language for high-level reactive network control. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 43–48. ACM, 2012.

[117] Bruno Lopes Alcantara Batista and Marcial Porto Fernandez. Ponderflow: A new policy specification language to sdn openflow-based networks. *International Journal on Advances in Networks and Services*, 7(3 & 4), 2014.

[118] Wonkyu Han, Hongxin Hu, and Gail-Joon Ahn. LPM: Layered policy management for software-defined networks. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 356–363. Springer, 2014.

[119] Eder J Scheid, Cristian C Machado, Ricardo L dos Santos, Alberto E Schaeffer-Filho, and Lisandro Z Granville. Policy-based dynamic service chaining in network functions virtualization. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 340–345. IEEE, 2016.

[120] Vijay Varadharajan, Kallol Karmakar, Uday Tupakula, and Michael Hitchens. A policy-based security architecture for software-defined networks. *IEEE Transactions on Information Forensics and Security*, 14(4):897–912, 2018.

[121] Nicolas Schnepf, Rĕmi Badonnel, Abdelkader Lahmadi, and Stephan Merz. Synaptic: A formal checker for sdn-based security policies. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–2. IEEE, 2018.

[122] Robert Craven, Jorge Lobo, Emil Lupu, Alessandra Russo, and Morris Sloman. Policy refinement: Decomposition and operationalization for dynamic domains. In *2011 7th International Conference on Network and Service Management*, pages 1–9. IEEE, 2011.

[123] Cristian Cleder Machado, Lisandro Zambenedetti Granville, Alberto Schaeffer-Filho, and Juliano Araujo Wickboldt. Towards SLA policy refinement for QoS management in software-defined networking. In *2014 IEEE 28th International Conference on AINA*, pages 397–404. IEEE, 2014.

[124] Cristian Cleder Machado, Juliano Araujo Wickboldt, Lisandro Zambenedetti Granville, and Alberto Schaeffer-Filho. Policy authoring for software-defined networking management. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 216–224. IEEE, 2015.

[125] Cristian Cleder Machado, Juliano Araujo Wickboldt, Lisandro Zambenedetti Granville, and Alberto Schaeffer-Filho. Arkham: an advanced refinement toolkit for handling service level agreements in software-defined networking. *Journal of Network and Computer Applications*, 90:1–16, 2017.

[126] Syed Noorulhassan Shirazi, Steven Simpson, Kanza Noor Syeda, Andreas Mauthe, and David Hutchison. Towards policy refinement for resilience management in cloud. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 260–266. IEEE, 2016.

[127] Gary N Stone, Bert Lundy, and Geoffrey G Xie. Network policy languages: a survey and a new approach. *IEEE network*, 15(1):10–21, 2001.

[128] Jiamei Liu and Qiaozhi Xu. Machine learning in software defined network. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 1114–1120. IEEE, 2019.

[129] Junfeng Xie, F Richard Yu, Tao Huang, Renchao Xie, Jiang Liu, Chenmeng Wang, and Yunjie Liu. A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(1):393–430, 2018.

[130] Anteneh A Gebremariam, Muhammad Usman, and Marwa 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)*, pages 545–549. IEEE, 2019.

[131] Majd Latah and Levent Toker. Artificial intelligence enabled software-defined networking: a comprehensive overview. *IET Networks*, 8(2):79–99, 2018.

[132] Majd Latah and Levent Toker. Application of artificial intelligence to software defined networking: A survey. *Indian Journal of Science and Technology*, 9(44), 2016.

[133] Junaid Qadir, Kok-Lim Alvin Yau, Muhammad Ali Imran, Qiang Ni, and Athanasios V Vasilakos. Ieee access special section editorial: Artificial intelligence enabled networking. *IEEE Access*, 3:3079–3082, 2015.

[134] Raouf Boutaba, Mohammad A Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M Caicedo. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1):16, 2018.

[135] AysŞe Rumeysa Mohammed, Shady A Mohammed, and Shervin Shirmohammadi. Machine learning and deep learning based traffic classification and prediction in software defined networking. In *2019 IEEE International Symposium on Measurements & Networking (M&N)*, pages 1–6. IEEE, 2019.

[136] Pu Wang, Shih-Chun Lin, and Min Luo. A framework for qos-aware traffic classification using semi-supervised machine learning in sdns. In *Services Computing (SCC), 2016 IEEE International Conference on*, pages 760–765. IEEE, 2016.

[137] Pedro Amaral, Joao Dinis, Paulo Pinto, Luis Bernardo, Joao Tavares, and Henrique S Mamede. Machine learning in software defined networks: data collection and traffic classification. In *Network Protocols (ICNP), 2016 IEEE 24th International Conference on*, pages 1–5. IEEE, 2016.

[138] Mohammad Reza Parsaei, Mohammad Javad Sobouti, Seyed Raouf Khayami, and Reza Javidan. Network traffic classification using machine learning techniques over software defined networks. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 8(7):220–225, 2017.

[139] Albert Mestres, Alberto Rodriguez-Natal, Josep Carner, Pere Barlet-Ros, Eduard Alarcón, Marc Solé, Victor Muntés-Mulero, David Meyer, Sharon Barkai, Mike J Hibbett, et al. Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review*, 47(3):2–10, 2017.

[140] Atiku Abubakar and Bernardi Pranggono. Machine learning based intrusion detection system for software defined networks. In *Emerging Security Technologies (EST), 2017 Seventh International Conference on*, pages 138–143. IEEE, 2017.

[141] Luca Boero, Mario Marchese, and Sandro Zappatore. Support vector machine meets software defined networking in ids domain. In *Teletraffic Congress (ITC 29), 2017 29th International*, volume 3, pages 25–30. IEEE, 2017.

[142] Saurav Nanda, Faheem Zafari, Casimer DeCusatis, Eric Wedaa, and Baijian Yang. Predicting network attack patterns in sdn using machine learning approach. In *Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on*, pages 167–172. IEEE, 2016.

[143] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. Deep learning approach for network intrusion detection in software defined networking. In *Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on*, pages 258–263. IEEE, 2016.

[144] Xuancheng Guo, Hui Lin, Zhiyang Li, and Min Peng. Deep reinforcement learning based qos-aware secure routing for sdn-iot. *IEEE Internet of Things Journal*, 2019.

[145] Tuba Uzakgider, Cihat Cetinkaya, and Muge Sayit. Learning-based approach for layered adaptive video streaming over sdn. *Computer Networks*, 92:357–368, 2015.

[146] Sandra Sendra, Albert Rego, Jaime Lloret, Jose Miguel Jimenez, and Oscar Romero. Including artificial intelligence in a routing protocol using software defined networks. In *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*, pages 670–674. IEEE, 2017.

[147] Shih-Chun Lin, Ian F Akyildiz, Pu Wang, and Min 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*, pages 25–33. IEEE, 2016.

[148] Md Billal Hossain and Jin Wei. Reinforcement learning-driven qos-aware intelligent routing for software-defined networks. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5. IEEE, 2019.

[149] Subham Kumar, Gaurang Bansal, and Virendra Singh Shekhawat. A machine learning approach for traffic flow provisioning in software defined networks. In *2020 International Conference on Information Networking (ICOIN)*, pages 602–607. IEEE, 2020.

[150] Giorgio Stampa, Marta Arias, David Sánchez-Charles, Victor Muntés-Mulero, and Albert Cabellos. A deep-reinforcement learning approach for software-defined networking routing optimization. *arXiv preprint arXiv:1709.07080*, 2017.

[151] Thuy TT Nguyen and Grenville Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials*, 10(4):56–76, 2008.

[152] George Apostolopoulos, Roch Guérin, Sanjay Kamat, and Satish K Tripathi. Quality of service based routing: A performance perspective. *ACM SIGCOMM Computer Communication Review*, 28(4):17–28, 1998.

[153] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[154] Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

[155] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[156] Joel Tropp, Anna C Gilbert, et al. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[157] Piet Van Mieghem, Fernando A Kuipers, Turgay Korkmaz, Marwan Krunz, Marília Curado, Edmundo Monteiro, Xavier Masip-Bruin, Josep Solé-Pareta, and Sergio Sánchez-López. Quality of service routing. In *Quality of Future Internet Services*, pages 80–117. Springer, 2003.

[158] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[159] Manish Parashar and Salim Hariri. Autonomic computing: An overview. In *International workshop on unconventional programming paradigms*, pages 257–269. Springer, 2004.

[160] Zaib Ullah, Fadi Al-Turjman, Leonardo Mostarda, and Roberto Gagliardi. Applications of artificial intelligence and machine learning in smart cities. *Computer Communications*, 154:313 – 323, 2020.

[161] Zaib Ullah, Fadi Al-Turjman, Uzair Moatasim, Leonardo Mostarda, and Roberto Gagliardi. Uavs joint optimization problems and machine learning to improve the 5g and beyond communication. *Computer Networks*, 182:107478, 2020.

[162] I. Comşa, 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*, pages 1–13, 2020.

[163] 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)*, pages 271–276, 2020.

[164] Gerald R Ash. *Traffic engineering and QoS optimization of integrated voice and data networks*. Elsevier, 2006.

[165] Andrew S Tanenbaum et al. *Computer networks*. New Jersey, 1996.

[166] Mansour J Karam and Fouad A Tobagi. On traffic types and service classes in the internet. In *Globecom'00-IEEE. Global Telecommunications Conference. Conference Record (Cat. No. 00CH37137)*, volume 1, pages 548–554. IEEE, 2000.

[167] Robert Braden, David Clark, Scott Shenker, et al. Integrated services in the internet architecture: an overview, 1994.

[168] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services, 1998.

[169] Amira Y Haikal, M Badawy, and Hesham A Ali. Towards internet qos provisioning based on generic distributed qos adaptive routing engine. *The Scientific World Journal*, 2014, 2014.

[170] Eleni Mykoniati, Charalampos Charalampous, Panos Georgatsos, Takis Damilatis, Danny Goderis, Panos Trimintzios, George Pavlou, and David Griffin. Admission control for providing QoS in DiffServ IP networks: the TEQUILA approach. *Communications Magazine, IEEE*, 41(1):38–44, 2003.

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

[172] Ke Li and Jitendra Malik. Learning to optimize. *International Conference on Learning Representations (ICLR)*, 2017.

[173] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[174] Marc Ponsen, Matthew E Taylor, and Karl Tuyls. Abstraction and generalization in reinforcement learning: A summary and framework. In *International Workshop on Adaptive and Learning Agents*, pages 1–32. Springer, 2009.

[175] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[176] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[177] 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.

[178] Mininet openflow virtual network. `http://mininet.org`. Accessed on Aug 1, 2020.

[179] Felipe Alencar, Marcelo Santos, Matheus Santana, and Stenio Fernandes. How software aging affects sdn: A view on the controllers. In *2014 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–6. IEEE, 2014.

[180] Iperf network testing tool for bandwidth measurements. `https://iperf.fr/`. Accessed on Aug 1, 2020.

[181] Ajay Tirumala. Iperf: The tcp/udp bandwidth measurement tool. *http://dast.nlanr.net/Projects/Iperf/*, 1999.

[182] Anees Shaikh, Jennifer Rexford, and Kang G. Shin. Evaluating the Impact of Stale Link State on Quality-of-Service Routing. *IEEE/ACM Transactions on Networking*, 9:162–176, 2001.

[183] Ferrel G Stremler. Introduction to communication systems. In *ics*, 1990.

[184] Alexei Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge & Management*, 14, 2019.

[185] CAIDA: Center for applied Internet data analysis. `http://www.caida.org/`. Accessed on Aug 1, 2020.

[186] Bo Zhou, Dan He, and Zhili Sun. Traffic modeling and prediction using arima/garch model. In *Modeling and Simulation Tools for Emerging Telecommunication Networks*, pages 101–121. Springer, 2006.

[187] Xin Yuan and Guang Yang. Empirical probability based qos routing. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 3, pages 1713–1717. IEEE, 2003.

[188] George Apostolopoulos, Roch Guerin, Sanjay Kamat, and Satish Tripathi. Improving qos routing performance under inaccurate link state information. In *Proceedings of the 16th International Teletraffic Congress*, pages 7–11. Edinburgh, UK: Elsevier Press, 1999.

[189] Xin Yuan, Wei Zheng, and Shiling Ding. A comparative study of qos routing schemes that tolerate imprecise state information. In *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*, pages 230–235. IEEE, 2002.

[190] Karol Kowalik and Martin Collier. ALCFRA - A robust routing algorithm which can tolerate imprecise network state information. *in Proceedings of 15th ITC Specialist Seminar, Wurzburg, Germany*, July 2002.

[191] Deepankar Medhi and Karthikeyan Ramasamy. *Network routing: algorithms, protocols, and architectures*. Morgan Kaufmann, 2010.

[192] Geoff Cumming and Robert Calin-Jageman. *Introduction to the new statistics: Estimation, open science, and beyond*. Routledge, 2016.

[193] Douglas G Altman and J Martin Bland. Standard deviations and standard errors. *Bmj*, 331(7521):903, 2005.

[194] Anilkumar Vishnoi, Rishabh Poddar, Vijay Mann, and Suparna Bhattacharya. Effective switch memory management in openflow networks. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, pages 177–188, 2014.

[195] Razvan Beuran. *Introduction to network emulation.* Pan Stanford, 2019.

[196] CpQD OpenFlow1.3 Software Switch. `http://cpqd.github.io/ofsoftswitch13/`. Accessed on Aug 1, 2020.

[197] Simon Knight, Hung X Nguyen, Nick Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 29(9):1765–1775, 2011.

[198] Ammar Muthanna, Abdelhamied A Ateya, Maria Makolkina, Anastasia Vybornova, Ekaterina Markova, Alexander Gogol, and Andrey Koucheryavy. Sdn multi-controller networks with load balanced. In *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, pages 1–6, 2018.

[199] Othmane Blial, Mouad Ben Mamoun, and Redouane Benaini. An overview on sdn architectures with multiple controllers. *Journal of Computer Networks and Communications*, 2016, 2016.

[200] Ostinato traffic generator tool. `https://ostinato.org/`. Accessed on Aug 1, 2020.

[201] Cisco Visual Networking Index. Forecast and Methodology, 2015–2020 White Paper, Cisco, 2016, 2016.

[202] Open vSwitch. `http://openvswitch.org`. Accessed on Aug 1, 2020.

[203] FFMPEG video and audio converter tool. `https://ffmpeg.org/`. Accessed on Aug 1, 2020.

[204] Andrew S Tannenbaum. *Computer networks.* Pearson Education India, 2002.

[205] VS Bagad and IA Dhotre. *Computer Networks-II.* Technical Publications, 2009.

[206] Syed Asad Hussain. *Active and programmable networks for adaptive architectures and services.* CRC Press, 2006.

[207] William James Dally and Brian Patrick Towles. *Principles and practices of interconnection networks.* Elsevier, 2004.

[208] Hongqiang Zhai, Xiang Chen, and Yuguang Fang. How well can the ieee 802.11 wireless lan support quality of service? *IEEE Transactions on Wireless Communications*, 4(6):3084–3094, 2005.

[209] IT 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.

[210] Umesh Lakshman and Lancy Lobo. *MPLS configuration on Cisco IOS software*. Cisco Press, 2005.

[211] Steve Pickavance Chris Lewis. *Implementing Quality of Service Over Cisco MPLS VPNs*. Cisco Press, 2006.

[212] Troubleshooting packet loss: How much is an acceptable amount? `https://www.vyopta.com/blog/video-conferencing/understanding-packet-loss/`. Accessed on Aug 1, 2020.

[213] Tim Szigeti, Christina Hattingh, Robert Barton, and Kenneth Briley Jr. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*. Cisco Press, 2013.

[214] Warren Matthews. Internet monitoring in the hep community. Technical report, Stanford Linear Accelerator Center, Menlo Park, CA (US), 1998.

[215] Farnaz Farid. Methodical evaluation of quality of service for heterogeneous networks. *PhD Thesis,*, 2015.

[216] Seung-Bum Lee, Gabriel-Miro Muntean, and Alan F Smeaton. Performance-aware replication of distributed pre-recorded iptv content. *IEEE Transactions on Broadcasting*, 55(2):516–526, 2009.