2020

QoE-Centric Control and Management of Multimedia Services in Software Defined and Virtualized Networks

Barakabitze, Alcardo Alex

http://hdl.handle.net/10026.1/15821

http://dx.doi.org/10.24382/1083
University of Plymouth

QoE-Centric Control and Management of Multimedia Services in Software Defined and Virtualized Networks

by

ALCARDO ALEX BARAKABITZE

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Engineering, Computing and Mathematics

March 2020

Abstract

Multimedia services consumption has increased tremendously since the deployment of 4G/LTE networks. Mobile video services (e.g., YouTube and Mobile TV) on smart devices are expected to continue to grow with the emergence and evolution of future networks such as 5G. The end user's demand for services with better quality from service providers has triggered a trend towards Quality of Experience (QoE) - centric network management through efficient utilization of network resources. However, existing network technologies are either unable to adapt to diverse changing network conditions, or limited in available resources. This has posed challenges to service providers for provisioning of QoE-centric multimedia services.

New networking solutions such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) can provide better solutions in terms of QoE control and management of multimedia services in emerging and future networks. The features of SDN, such as adaptability, programmability and cost effectiveness make it suitable for bandwidth intensive multimedia applications such as live video streaming, 3D/HD video and video gaming. However, the delivery of multimedia services over SDN/NFV networks to achieve optimized QoE, and the overall QoE-centric network resource management remain an open question especially in the advent development of future softwarized networks.

The work in this thesis intends to investigate, design and develop novel approaches for QoE-centric control and management of multimedia services (with a focus on video streaming services) over software defined and virtualized networks. First, a video quality management scheme based on the traffic intensity under Dynamic Adaptive Video Streaming over HTTP (DASH) using SDN is developed. The proposed scheme can mitigate virtual port queue congestion which may cause buffering or stalling events during video streaming, thus, reducing the video quality.

A QoE-driven resource allocation mechanism is designed and developed for improving the end user's QoE for video streaming services. The aim of this approach is to find the best combination of network node functions that can provide an optimized QoE level to end-users through network node cooperation. Furthermore, a novel QoE-centric management scheme is proposed and developed, which utilizes Multipath TCP (MPTCP) and Segment Routing (SR) to enhance QoE for video streaming services over SDN/NFV-based networks. The goal of this strategy is to enable service providers to route network traffic through multiple disjointed bandwidth-satisfying paths and meet specific service QoE guarantees to the end-users. Extensive experiments demonstrated that the proposed schemes in this work improve the video quality significantly compared with the state-of-the-art approaches. The thesis further proposes the path protections and link-failure free MPTCP/SR-based architecture that increases Survivability , resilience,

availability and robustness of future networks. The proposed path protection and dynamic link recovery scheme achieves a minimum time to recover from a failed link and avoids link congestion in softwarized networks.

Contents

List of Figures

List of Tables

Acknowledgements

This work is the culmination of a journey that started in September 2015. Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

First and foremost, I would like to express my sincere gratitude to my Director of Studies Prof. Lingfen Sun for the continuous support, patience, motivation, and immense knowledge she gave to me since when I started my PhD study. Her professional guidance helped me in all the time of research and writing of this thesis. I appreciate all her contributions of time, encouragement and ideas to make my PhD experience productive and stimulating. The joy and enthusiasm she has for her research will remain to be contagious and motivational for me throughout the journey of my research career. It has been an enormous privilege to learn from her expertise and leadership and I am also thankful for the excellent example she has provided to me of working very hard. My numerous quality papers that I was able to publish during my four years of studying towards PhD would not have been possible without her and for that I am truly grateful.

I am also deeply in debt to my PhD supervisors, Prof. Emmanuel Ifeachor and Dr. Is-Haka Mkwawa who always had a door open when I needed someone to discuss and get feedback from. Their insights and comments have shaped this PhD thesis. A very special thanks to them for their advice and support during my four years of pursuing my PhD. I have hugely benefitted from the collaborations and discussions with them. I collectively thank my supervisory team for inspirational guidance, support and constructive discussions with me regarding the experiments that helped me to publish papers in different conferences and journals. I am truly grateful to Prof. Lingfen Sun, Prof.Emmanuel Ifeachor and Dr.Is-Haka Mkwawa for helping me in whatever way they could during this challenging period. I would also like to thank all the reviewers who reviewed my papers. Without their constructive comments and suggestions I would not have been able to raise the standard of my publications.

A particular mention also goes to all members of Signal Processing Multimedia Communications (SPMC) Research group: Dr Emmanuel Jammeh for any kind of help, support and constructive discussions he gave to me in the lab. It has been my great pleasure to have been working in the SPMC group. I thank all of my fellow lab mates from SPMC for the stimulating discussions and for all the fun moments we have had together in the last four years. It has really been a privilege to be part of such an amazing team.

I gratefully acknowledge the funding received towards my PhD from the EU Horizon 2020 Research and Innovation Programme under Marie Sklodowska-Curie Innovative Training Networks (MSCA-ITN-2014-ETN), Grant Agreement No.643072, Network QoE-NET. My thanks also go out to the support I received

from the ITU - T Study 12 (Network performance, QoS, and QoE) when I was undertaking my Internship as an Early Carer Researcher (ESR). I am especially grateful to Martin Adolph for all the discussions we had in the field of performance, QoS, and QoE through ITU-T Study 12 meetings.

Infinite thanks to my parents, Alex Barakabitze and Leonia Francis Nzika, my sisters and brothers, Josaphine, Alfred, Abely and Bibiana for all the support and unconditional love over the years. I would also like to thank my wife, Lilian Alcardo Barakabitze for her encouragement, support and unconditional love during this amazing journey. Lastly, this thesis is dedicated to our beautiful two children, Alvin and Niah for their motivation, support, endless love and for bearing with me in my PhD study journey. Not only they have given me the strength, stability, and motivation to finish this work, they have also made my life brighter and lighter with their love.

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

Relevant scientific seminars and conferences were regularly attended at which work was often presented:

Word count for the main body of this thesis: **51.300 words**

Signed: _____

Date: 24th February 2020

Chapter 1

Introduction

Multimedia services consumption has increased tremendously over the past years and is expected to continue to grow even more over the next years. According to the latest Cisco Visual Networking Index (VNI) Forecast [7], over three-fourths of the world's data traffic is going to be video by 2021. Mobile-connected devices are estimated to be 11.6 billion including Device to Device (D2D) and Machine to Machine (M2M) communications and therefore exceeding the world's projected population of 7.8 billion by 2021. The exponential growth of video streaming services (e.g., YouTube and Mobile TV) on smart devices has triggered and introduced new revenue potential for telecom operators and service providers.

This fact, coupled with the increasing numbers of data consuming devices which run enhanced applications such as live video streaming, social networking, 3D/HD video, video gaming and Virtual Reality pose a challenge to mobile and service providers in terms of management of multimedia networks as end-users become accustomed to more resource demanding services with better quality. Fixed and mobile networks today are converging towards the future telecommunication domain (e.g., 5G networks) through the integration of existing and new wireless technologies. Such integration is set to provide access to any service with better quality to the end-users through a reliable and cost effective communication, over any medium and across multi-operator domains using different networking technologies [8]. Therefore, future networks should enable end-users to access multimedia services with high Quality of Experience (QoE) from service providers at anytime, anywhere without limitations on technology and medium. In order to meet these requirements, new architectures and intelligent schemes for QoE control

and management of future multimedia services have to be designed and developed.

1.1 Research Challenges and Motivations

The end-users' demand for services with better quality from service providers has triggered the trend of traditional and future networks towards using QoE in network management through an efficient utilization of network resources. In order to meet end-users' QoE requirements and their expectations, several QoE issues (e.g., QoE monitoring, QoE control and management) pose challenges to service providers [9]. Such issues are contributed by existing technologies which are either unable to adapt to diverse changing network conditions, or limited in the available resources. In addition, existing QoE adaptation approaches limit the network ability to provide intelligent and efficient solutions such that they only react when a problem occurs and therefore resulting in sub-optimal network performance. These challenges necessitate harmonizing the integration of advanced intelligence architectures in future networks with the development of intelligent QoE-aware adaptation mechanisms. Therefore QoE enabled functions, such as QoE-centric routing, admission QoE control, resource allocation and server selection mechanisms in future networks should be adaptive to changing network conditions to improve end-users' QoE.

Today, end-users are accustomed to more resource demanding services with better quality from Service Providers (SP) [10]. However, achieving good Quality of Experience (QoE) is a challenging task because of different clients devices/requests patterns, changing media contents and varying transmission/network conditions. Great efforts from both academia and industry have been made to optimize the video content delivery chain and enhance end users' QoE. Some of the common mechanisms used for improving end-users' QoE are either based on network optimization (e.g. QoE-driven network resource allocation and QoE-driven routing) or based on client-server request/response (e.g. client-driven adaptive video streaming) and QoE-based routing [11, 12, 9].

Despite these efforts, QoE management remains a challenging task due to many issues [9, 13]. Firstly, the variability of network resources, and unstable nature of wireless channels and the characteristics of fixed/mobile networks in heterogeneous environments, for example, crowded areas such as train, stadium and shopping malls. This kind of environments requires continuous adaptation of network resources allocation. Secondly, the emergence of new services (e.g., video gaming and virtual/augmented reality (VR/AR)),

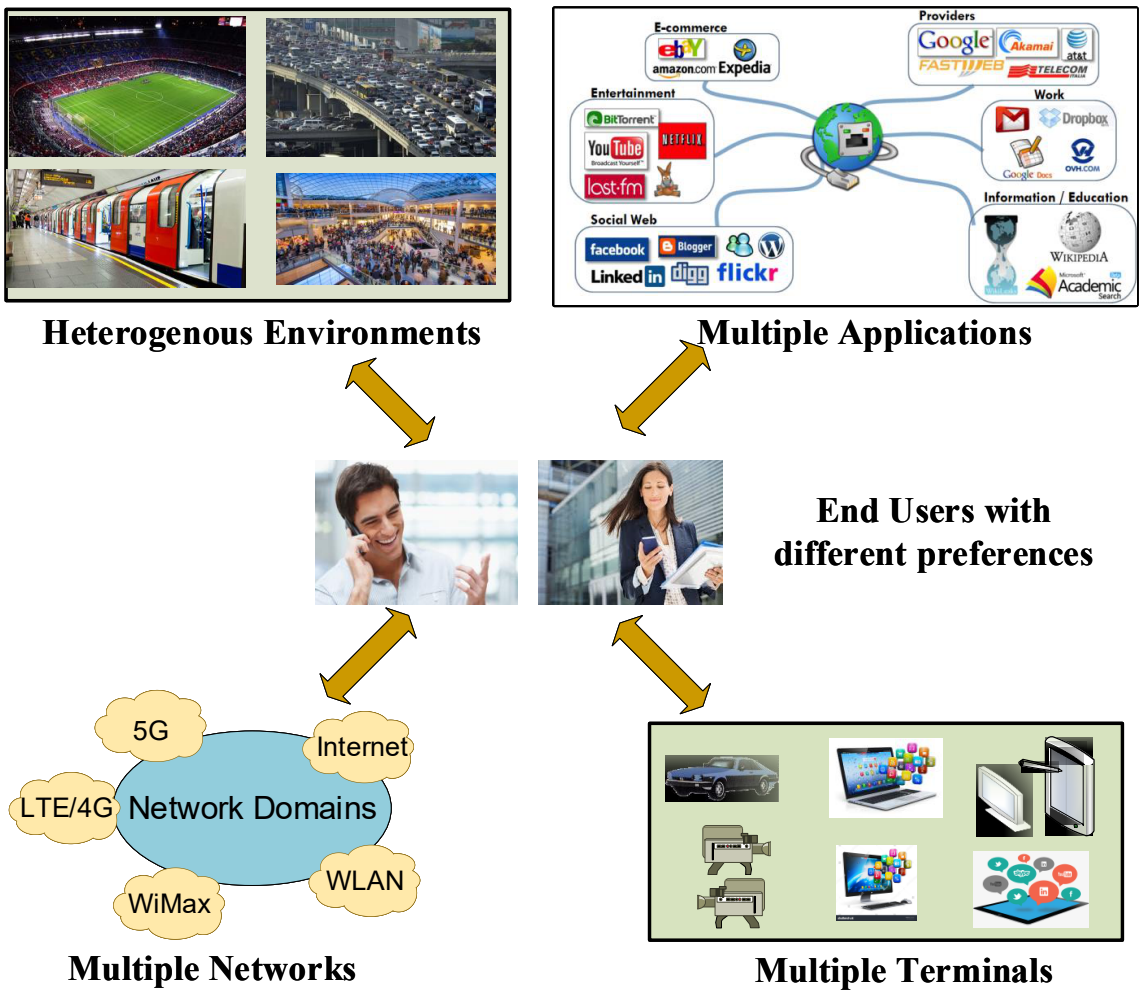

Figure 1.1: QoE management challenges in future networks.

the diversity of contexts of use, the users' expectations combined with the operational cost optimization by mobile and service providers lead QoE management into challenging zones of the research field. Thirdly, considering a variety of networks (e.g., fixed and mobile), different measurement and assessment methods are required to be employed for QoE management considering resources constraints. Fourth, the popularity and fast growth of the multimedia services over the Internet, the heterogeneity of end-user devices with different capabilities (e.g., screen size, computational power/resources, and storage capabilities) pose even more challenges regarding resource allocation among users with different QoE preferences. Fig. 1.1 summarizes the QoE management challenges in current and future IP-based networks.

Increased user demand and expectations of services with excellent quality have triggered telecom operators to upgrade their systems and invest in new cutting-edge network softwarization paradigms such as Software Defined Networking (SDN) [14], Network Function Virtualization (NFV) [15], Multi-access Edge Computing (MEC) [16] and Cloud/Fog

Computing (C/FoC). This transformation and the upgrade of their systems are also driven by the mounting pressure of new emerging use cases, ranging from full ultra-high definition video resolutions (4K/8K), network-controlled D2D communications, Machine Type Communication (MTC) and Massive Internet of Things (MIoT).

In such an agile and flexible environment, it is important to consider new solutions, such as the separation of user and control planes, and possibly, re-define the boundaries between the network domains (for example, radio access network and the core network). Therefore, there is a need for autonomic network management platforms that can guarantee the end users' QoE, especially in heterogeneous environments. Towards this end, new paradigms such as SDN and NFV have been identified as critical technologies for enabling future network control to be programmable, centrally manageable, adaptable and cost-effective. SDN and NFV are considered to be suitable for applications such as video streaming [17, 18, 19, 20, 21, 22, 23, 24]. Indeed, SDN and NFV may enable network management to be automated and ensure that the end-user's QoS/QoE requirements as well as Experience Level Agreements (ELAs)[1] are fulfilled in the heterogeneous environments [25].

More importantly, SDN and NFV can provide an end-to-end resource, infrastructure and service orchestration across a multi-programmable domain that belongs to different operators or service providers. To this end, investigations are ongoing as to what extent future networks and different supporting technologies can be software-configurable and to what extent software platforms can be hardware-agnostic. Aligned with the on-going SDN and NFV research activities by some standardization bodies, there is an urgent need to study and explore the adoption of SDN and NFV on how they introduce the significant changes on the operation, management, and delivery of QoE-aware services in the context of emerging and future networks.

1.2 Research Aims and Objectives

The main aim of this thesis is to investigate and develop Quality of Experience (QoE) - centric control and management schemes for streaming video services over software defined and virtualized networks. Specific objectives of the research work are to:

[1]Experience Level Agreements (ELAs): Indicates a QoE-enabled counter piece to traditional QoS-based that conveys the performance of the service in terms of QoE. ELAs establish a common understanding of an end-user experience on the quality levels whiling using the service

- Investigate comprehensively the QoE-centric control and management for HTTP adaptive video streaming using SDN/NFV. The aim is to overcome future challenges and achive an optimized QoE to the end users with different client devices/request patterns at varying transmission/network conditions.

- Investigate a QoE-based multimedia flow routing mechanism using traffic intensity over SDN/NFV. The proposed scheme can mitigate virtual port queue congestion and reduce buffering or stalling events during video streaming.

- Design, formulate and implement the QoE-Centric multipath routing approach for multimedia services using SDN. The aim is to utilize multiple disjointed paths to improve network resources utilization and end-users' QoE for multimedia services delivery in future softwarized networks.

- Design a novel QoE-driven SDN based resource allocation mechanism for future SDN/NFV-based networks. A combination of network node functions are used to provide an optimized QoE level to end-users through node cooperation in softwarized networks.

- Investigate and design a QoE-centric management of multimedia services using MPTCP and Segment Routing (SR) in softwarized networks. A QoE-based multipath routing and video quality optimization algorithm is proposed to provide an efficient orchestration, QoE control and management of future multimedia (video) services.

- Investigate, design and implement a multipath protections and link- failure free MPTCP/SR-based architecture that increases survavibility, resilience, availability and robustness of 5G networks.

- Perform extensive experiments and quality evaluations from real-time multimedia transmissions over the developed softwarized network platform.

1.3 Summary of Thesis Contributions

This thesis investigates broadly the QoE-centric control and management of video streaming services in software defined and virtualized networks. It is important to mention that, apart from addressing the QoE-centric control and management, another contribution of this thesis include the development of an SDN/NFV-based experimental platform for

conducting video streaming experiments. Specifically, the SDN controller was extended by adding QoE management module, Multipath TCP (MPTCP) flow routing module and SR module as presented in Chapters 5 and 6. Figure 1.2 illustrates the approach that was followed in this thesis by providing a relationship between chapters, publications and its contributions in the field. Specifically, the main contributions of this thesis are outlined and summarized in the following five (5) areas.

1.3.1 QoE-based Multimedia Flow Routing Mechanisms using SDN/NFV

A video quality management scheme over an SDN architecture based on traffic intensity is proposed to minimize virtual port queue congestion which may cause buffering or stalling events during video streaming, thus, reduce the video quality. The QoE-based multimedia flow routing approach in SDN/NFV is designed to (a) investigate the performance of the control plane particularly the OpenFlow protocol, and (b) avoid the queuing overflow that can negatively affect the video quality of the transmitted video from server to an end-user. The work has been contributed to the research community following the publication achieved in [26]. The contributions in this work are described in Chapter 3. As part of my contributions in this work, I conducted video streaming experiments over SDN and analyzed the results.

1.3.2 A QoE-Driven SDN based Resource Allocation Algorithm in Future Networks

With the continuous growth of multimedia applications in future networks such as 5G, enabling the dynamic configuration and resource allocation in SDN-based networks can be significant and an outstanding approach of improving the end-users' QoE. In this regards, this work applied network softwarization and virtualization technologies specifically SDN and NFV to propose a novel QoE-driven resource allocation mechanism that assigns tasks dynamically to virtual network nodes in order to achieve an optimized end-to-end quality. The aim of the proposed approach is to find the combination of network node functions that can provide an optimized QoE level to end-users through node cooperation. The author considered multimedia service as a collection of tasks where neighbor nodes negotiate the assignment of these tasks by considering the end-users' perceived video quality. The work has been contributed to the research community

following the publication achieved in [27]. The contributions in this work are described in Chapter 4. My contributions in this work is on experimental part, evaluation of the results and algorithm implementation in SDN.

1.3.3 A Novel QoE-Centric Multipath Routing Algorithm for Video Streaming using SDN

A multiflow approach is employed in this contributions where multiple disjointed paths are used in SDN-based networks to improve network resources utilization and end-users' QoE for delivering video streaming services. Specifically, the work provides the first implementation mechanisms of MPTCP and SR for traffic management in software defined and virtualized networks. DASH is employed to test the performance and effectiveness of the proposed algorithm. It is important to mention that, this strategy is compared with the traditional TCP approach. The Stream Control Transmission Protocol (SCTP) which operates at the transport layer is not compared because it serves a role similar to TCP and UDP [28]. The contribution under this work proposes a novel QoE-aware SDN-based MPTCP/SR framework to enhance the QoE for video streaming services delivery in future networks. In the proposed framework, the POX controller was extended by introducing three modules namely, (a) MPTCP-Flow Manager, (b) QoE management, and (c) Segment Routing (SR) Module. The work has been contributed to the research community following the publication achieved in [29]. The contributions in this work are described in Chapter 5.

1.3.4 QoE Control and Management of Video Streaming Services in Softwarized Networks

This contribution provides the first original practical Traffic Engineering (TE) mechanisms where MPTCP and SR are further employed in softwarized networks to facilitate efficient transfer of large amount of multimedia applications between end-points. The chapter extends the contribution four (4) described above, where MPTCP and SR were utilized to improve video quality and the system performance for video streaming services. In order to improve the video quality, the Multi-flow commodity and Constrained Shortest Path Model (MCSPM) is applied to choose important intermediate nodes and perform source routing using SR paradigm. As a key contribution of the thesis, a novel QoE-aware

MPTCP/SR-enabled architecture in softwarized networks is proposed to provide an efficient orchestration, QoE control and management of future multimedia services. The work has been contributed to the research community following the publication achieved in [30]. The contributions in this work are described in Chapter 6.

1.3.5 Multipath Protections and Dynamic Link Recovery in Softwarized 5G Networks using Segment Routing

In this work, a multipath protection and link-failure free MPTCP/SR-based SDN/NFV architecture that increases survivability, resilience, availability of services in 5G networks is proposed. The system model and a multiPath protection and dynamic link- failure free algorithm called *"PathReLief"* is proposed to minimize the failure recovery time and avoids link congestion in MPTCP/SR softwarized 5G networks. To demonstrate the effectiveness of the proposed approach, the performance of the proposed algorithm is compared with the conventional topology discovery mechanisms for link/node failures in POX and OpenDaylight controllers. The work has been contributed to the research community following the publication achieved in [31]. The contributions in this work are described in Chapter 6.

1.4 Overview of Publications and Author Contributions

It is important to mention that, the thesis and the work described herein is covered in the following peer-reviewed publications and submitted papers. The papers cover technical and novel research-based position contributions in the QoE management as well as future networks domain.

TC1 : **A.A.Barakabitze**, L.Sun, IH Mkwawa and E.Ifeachor. "A Novel QoE-Aware SDN-enabled, NFV-based Management Architecture for Future Multimedia Applications on 5G Systems", in Eighth International Conference on Quality of Multimedia Experience (QoMEX), June, 2016, Lisbon, Portugal

TC2 :IH Mkwawa, **A.A.Barakabitze**, L.Sun. Video Quality Management over the Software Defined Networking, in IEEE International Symposium on Multimedia (ISM), December, 2016, San Jose, CA, USA.

TC3 :E. Grigoriou, **A.A.Barakabitze**, L.Atzori, L.Sun and V.Pilloni, "An SDN-approach for QoE Management of Future Multimedia Services using Resource Allocations", IEEE International Conference on Communications, May, 2017, Paris, France.

TC4 :**A.A.Barakabitze**, L.Sun, IH Mkwawa, E.Ifeachor, "A Novel QoE-Centric SDN-Multipath Routing Approach for Future Multimedia Services in 5G Networks", 2018 IEEE International Conference on Communications, May 2018, Kansas City, MO, USA.

TC5 :**A.A.Barakabitze**, IH Mkwawa, L.Sun, E.Ifeachor, "QualitySDN: Improving Video Quality using MPTCP and Segment Routing in SDN/NFV", IEEE International Conference on Network Softwarization (NetSoft), June 2018 - Montreal, Canada.

TC6 **A.A, Barakabitze**, L.Sun, IH Mkwawa and E.Ifeachor. "MultiPath Protections and Dynamic Link Recovery in Softwarized 5G Networks using Segment Routing", IEEE GLOBECOM Workshop on Future Internet Architecture, Technologies and Services for 2030 and beyond, December 2019, Waikoloa, HI, USA.

1.4.1 Journal Papers

TC7 **A.A, Barakabitze**, IH, Mkwawa, L, Sun and E, Ifeachor, "QoEMultiSDN: QoE Management of Multimedia Services using MPTCP and SR in SDN/NFV-based Networks", ACM Transactions on multimedia computing communications and applications.

PC1 **A.A Barakabitze**, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. Martini, L. Atzori and S. Moller, "QoE Management of Multimedia Services in Future Networks: A Tutorial and Survey", IEEE Communication Surveys and Tutorials, pp. 1-1, November, 2019.

PC2 **A. Barakabitze**, A. Arslan, M. Rashid, and H. Andrew, "5G Network Slicing using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges," Computer Networks, vol. 167, pp. 1-40, February 2020.

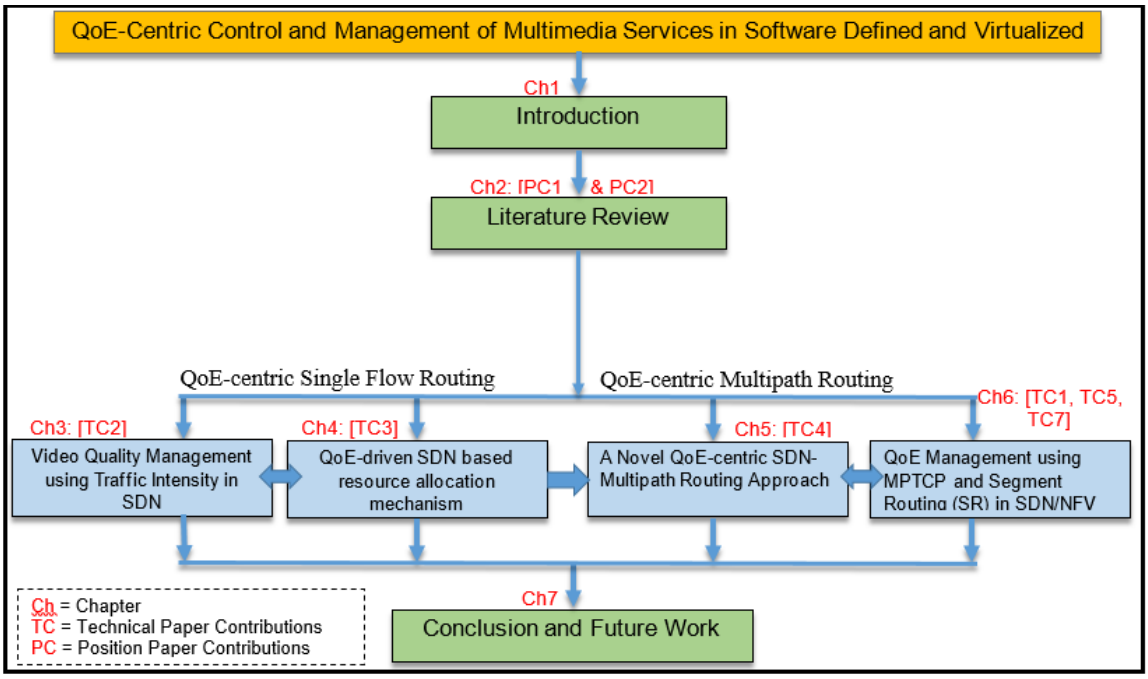

Figure 1.2: Organization and structural contributions of this thesis

1.5 Thesis Outline

The Organization and structural contributions of this thesis are illustrated in Figure 1.2. The studies covered in this thesis are indicated by chapters which form sections as visualized using boxes. The relationship between the sections are indicated using top-down arrows. The green boxes indicate chapters that cover the introduction, literature review and conclusion and future work while blue boxes indicate the major contributions of this thesis in the research field. In the end of each chapter, a summary of the work carried out and/or the lessons learned are provided. In the following, the organization and the overall structure of this thesis is described. After the thesis introduction, Chapter 2 presents a comprehensive description of literature review focusing on four main areas, listed as below. Section 2.2 provides highlights on the concept of QoE management for multimedia streaming services followed by a description of HTTP Adaptive Streaming (HAS) solutions including the Server and Network Assisted DASH (SAND) in Section 2.3. Section 2.4 presents the background information regarding network softwarization and virtualization technologies (e.g., SDN, NFV). Section 2.5 provides a detailed state-of-the-art regarding QoE management for HTTP adaptive video streaming using SDN/NFV.

Chapter 3 investigates the video quality management based on queue mechanisms and QoE/QoS Policy in SDN. In order to understand the traffic intensity approach, section 3.1

describes the introduction of queue mechanisms and QoE/QoS policies in SDN. Section 3.2 presents related work followed by a description of the proposed video quality management algorithm in section 3.3. A performance evaluation and results of the proposed scheme for DASH is given in Section 3.4. Section 3.5 provides lessons learned and a summary of this chapter.

Chapter 4 provides a novel QoE-driven resource allocation strategy that dynamically assign tasks to virtual network nodes in order to achieve an optimized end-to-end quality. Section 4.1 presents related work. Section 4.2 provides an illustration of a system overview and task assignment model followed by the description of the network model and a utility function in section 4.3. Section 4.4 presents the QoE-driven resource allocation and task assignment algorithm. Section 4.5 presents the performance evaluation and analysis of the results. Section 4.6 summarizes this chapter.

Chapter 5 explore the utilization of MPTCP and SR in SDN-based networks to improve the end-user's QoE for delivering video streaming services in future networks. Section 5.1 presents related work regarding multimedia transmission using MPTCP over SDN followed by a description of basic operation and principles of MPTCP in section 5.2. Section 5.3 provides the video adaptive streaming concept of MPTCP over SDN while section 5.4 presents the traffic engineering with segment routing in SDN. Section 5.5 presents a QoE-aware MPTCP SDN-based SR adaptation framework. Section 5.6 presents the proposed QoE-Centric Multipath Routing Algorithm (QoMRA) over SDN. Section 5.7 provides an experimental results and evaluation of the proposed QoE-aware MPTCP/SR-based algorithm. Section 5.8 summarizes the contributions of this chapter.

Chapter 6 presents a novel QoE-based multipath and video quality optimization algorithm called **"QoEMuSoRo"** that forwards traffic using SR paradigms over softwarized networks. In addition, Path Protection and Recovery of Link Failure (PathReLieF) algorithms are proposed to achieve dynamic route selection and link/node failure recovery over softwarized networks. Section 6.1 provides related work while section 6.2 provides a novel concept called "*QoE-softwarization"* in SDN and NFV. Section 6.3 presents the considerations of MPTCP over softwarized networks. Section 6.4 describes path protection and dynamic link recovery mechanisms with SR in softwarized networks. Section 6.5 provides the Service Function Chaining (SFC) strategies using SR in future softwarized network while section 6.6 provides a novel QoE-aware MPTCP and SR-based approach

over softwarized infrastructures leveraging the integration of SDN and NFV. Based on the concept from graph theory, section 6.7 presents a formulation of a system model and the proposed QoE-based multipath source routing algorithm. Section 6.8 presents the performance and evaluation of the proposed SDN/NFV system. Finally, section 6.9 summarizes this chapter.

Chapter 2

Literature Review

2.1 Introduction

The traditional best effort delivery paradigms have resulted into inadequate and uneconomical ways of providing the required level of end-users' QoE. Efficient application and network management for future networks (e.g., 5G) has the potential to deliver services enriched with high QoE to the end-users at a minimum cost. The context of QoE for future 5G networks takes into considerations the management perspectives on performance and quality in terms of networked communications through QoE aware service delivery systems. The main goal of mobile operators in implementing QoE management mechanisms is to maintain a maximized end-users' QoE in the delivered services while minimizing cost through an efficient allocation of available network resources [9]. With the increasing multimedia applications that require increased resources (e.g., high bandwidth, higher link quality) mobile operators have mostly been considering the QoE-driven resource management and QoE-driven service adaptation solutions as an important optimization approach for QoE management. Different QoE management mechanisms and frameworks for QoE optimization in wireless/wired environment have been proposed in the literature. Such mechanisms as we explain later use different approaches (e.g., differ in the chosen location, network or user-oriented) and parameters to be controlled as well as utilizing different network technologies and applications.

2.2 QoE Management for Multimedia Streaming Services

2.2.1 Quality of Exeprience (QoE): Definition

The Quality of Service (QoS) measurements using network parameters (e.g., bandwidth, packet loss, delay and jitter) have been considered for many years to define the level of satisfaction/performance of a service. While QoS indicates the level of performance of a service, it does not necessarily quantify the users' experience and level of satisfaction. Therefore, in recent years, QoE based measurement, in addition to QoS metrics have been widely used in industry to assess the quality of multimedia services. QoE considers the user's subjectivity towards a specific service. QoE is defined as *"the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state"* [32], [33]. The understanding of the users' expectations and experiences from a service is then vital for the success of a service.

2.2.2 QoE Monitoring and Measurement

The process of managing and optimizing the end user's QoE requires the knowledge regarding the root cause of QoE degradation or unsatisfactory QoE levels. In that aspect, relevant information and data related to terminal capabilities (e.g., screen size, display performance), application/service specific information and its quantification, QoE-related information inside the network have to be monitored, collected and measured [34]. The data and information or quality parameters can be collected from client devices or network elements using monitoring probes. With the advancement of virtualization and data management capabilities, QoE-awareness along the service delivery chain can be collected using virtualized probes [35]. The collected network-level Key Performance Indicators (KPIs, e.g., throughput, packet loss, delay) or user-level service/application specific Key Quality Indicators (KQI) (e.g., frame rate, video resolution, service usability, and reliability) provide inputs for QoE estimation models [36], [37].

2.2.3 QoE Optimization and Control of Multimedia Services

The QoE management of multimedia services involves continuous optimization and dynamic delivery control mechanisms. One of the ultimate goals of QoE management is the maximization of the end user's QoE level through the efficient allocation of available network resources. For example, a joint optimization approach of network resource allocation and video quality adaptation that fairly maximize video clients' QoE is given in [38]. Minimizing energy consumption especially in the context of mobile services has been another important objective considered when optimizing the end user's QoE [39]. Tao *et al.* [40] propose an energy-efficient video QoE optimization solution for DASH over wireless networks. Bouten *et al.* in [41] and [42] present in-network quality optimization agents,

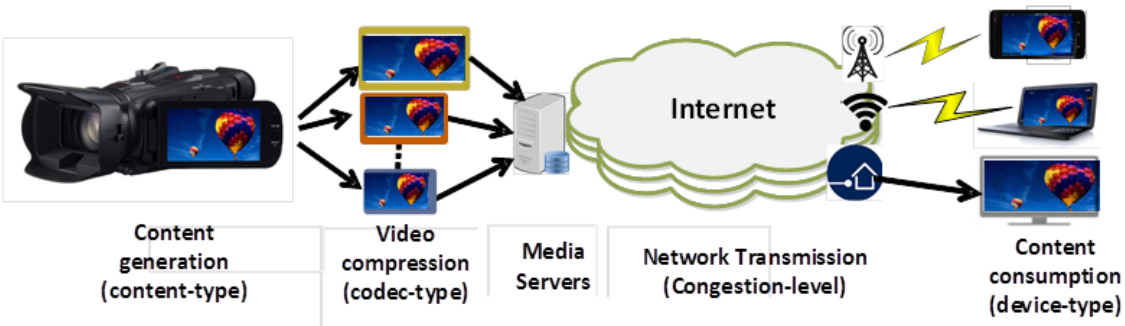

Figure 2.1: Multimedia streaming chain.

which monitor the available throughput using sampling-based measurement techniques. That way, the quality of each client is optimized based on a HAS QoE metric. Triki *et al.*[43] propose a dynamic closed-loop QoE optimization for video adaptation and delivery while the characterization of buffer starvation analysis with the aim of optimizing the end user's QoE is proposed by Xu *et al.* in [44].

However, considering the multimedia delivery chain as illustrated in Fig. 2.1, QoE optimization and control is a challenging task due to many issues including the heterogeneity of multimedia-capable user's devices. As stated in [9], the main challenges that arise with regards to QoE optimization and control may be summarized in the answers to the following four questions: (1) what key quality parameters to optimize and control? (2) where to control? (3) when to perform QoE optimization and control (e.g., during the service, that is, on-line control or in an off-line fashion)? [21] and (4) how often to control and optimize QoE? Answering the second question, for example, would consider optimizing QoE at various points during the video streaming process, for example at the

client, server and the network side.

2.3 Multimedia Streaming Services over the Internet

With the growth of streaming video traffic, it has become imperative to exploit various factors along the multimedia delivery chain to optimize the video service delivery by considering the end user QoE. This section presents a comprehensive discussion on HTTP adaptive streaming solutions over the Internet. Specifically, it highlights the server and client side optimization of video streaming including Dynamic Adaptive Streaming over HTTP (DASH) as well as SAND. Furthermore, key issues and challenges of HTTP adaptive video streaming are discussed. This is followed by a tutorial description of network softwarization technologies including SDN, NFV, network hypervisors and virtual machines. Moreover,this work provides the state-of-the-art regarding QoE management for HTTP adaptive video streaming using SDN/NFV. This thesis will build on the concepts presented in this section to investigate, design and develop novel approaches for QoE control and management of multimedia streaming services in future networks.

2.3.1 HTTP Adaptive Streaming (HAS) Solutions

The majority of Internet video traffic today is delivered via HTTP-based adaptive streaming (HAS), an approach that aims to provide a smooth streaming session for the users by keeping the highest possible video quality. The advantages of HAS include: (a) providing a reliable transmission, (b) cache infrastructure reuse capability, and (c) it can run over HTTP. Based on these benefits, HAS has been adopted commercially by almost all major vendors including the three biggest Over-The-Top (OTT) companies as shown in Table 2.1. HAS has been widely used in OTT video services such as Netflix and YouTube [45] as the de-facto standard for adaptive streaming solutions. The underlying logic is common in all these implementations with some differences in the manifest file, recommended segment size (See Table 2.1). HAS solutions use reliable delivery mechanisms such as TCP and very recently, Quick UDP Internet Connections (QUIC) [46]. However, due to the disparity of these proprietary HAS solutions and the media formats, the 3GPP in close collaboration with MPEG has developed a DASH standard [47]. As of today, DASH specifications consist of seven parts namely: (1) Media presentation description and segment formats, (2) Reference software and conformance bitstreams for DASH, (3) Implementation guidelines,

(4) Format Independent Segment encryption and authentication, (5) Server and network assisted DASH (SAND) [1], [48], (6) DASH with Server Push and WebSockets, and (7) Delivery of Common Media Application Format (CMAF)[1] content with DASH.

Table 2.1: A comparison of HTTP adaptive streaming solutions

HAS Category	Company	Video Codec	Segment Length (sec)	Data Description	Format
Microsoft Smooth Streaming [49]	Microsoft Corporation	H.264, VC-1	2	Manifest (XML)	fMP4
Apple HTTP Live Streaming (HLS) [50]	Apple Inc.	H.264	10	Playlist file (M3U8)	M2TS, *.ts files
Adobe HTTP Dynamic Streaming (HDS) [51]	Adobe Systems Inc.	H.264, VP6	2 - 5	Manifest (F4M)	fMP4
MPEG-DASH [47]	Standard	Any	Not specified	MPD files (XML)	MP4 or M2TS
3GP-DASH [52]	Standard	H.264	Not specified	MPD files (XML)	3GPP File Format

M2TS = MPEG-2 Transport Stream; fMP4 is a fragmented MP4; MPD = Media Presentation Description

Fig. 2.2 illustrates the concept of DASH assuming a throughput-based rate adaptation method. The DASH server contains video contents that are encoded for example with H.264 codecs and segmented based on GPAC MP4Box [53]. The video contents on the server are made up of the Media Presentation Description (MPD) and segments. The MPD describes a manifest of the available video contents, their URL addresses, characteristics (e.g., video periods, adaptation sets and various representations) and other metadata that may be needed by the clients. The video segments contain the actual multimedia bitstreams that the DASH client plays in single or multiple files in the form of chunks. The delivery of video segments, the client behavior for fetching, adaptation heuristics, and playing content are based on the QoE prediction and segment-aware algorithms that consider the granularity of the video [54].

The DASH client runs the DASH.js [55] that can be embedded with QoE prediction, buffer filling and bitrate guidance algorithms. These algorithms enable a DASH client to learn about the media-content availability, their media types, video resolutions, minimum and maximum bandwidths. It can be observed that the client, based on its network condition, adapts the quality of the video to provide a smooth streaming experience to the end user [54]. The video file is encoded at different representation levels (spatial/temporal/quality) and then divided into chunks (also referred to as segments) of equal duration, which are then stored on a server. When the client makes the first request for the video file, the server sends the corresponding manifest file (e.g., .mpd, .m3u8) which consists of the details about the video file such as duration, segment size, representation levels or codec type [11].

[1]CMAF is a new media streaming format whose standardization is being led within MPEG by Microsoft and Apple, with a view to making it both cheaper and easier to deliver OTT video streams to multiple devices.

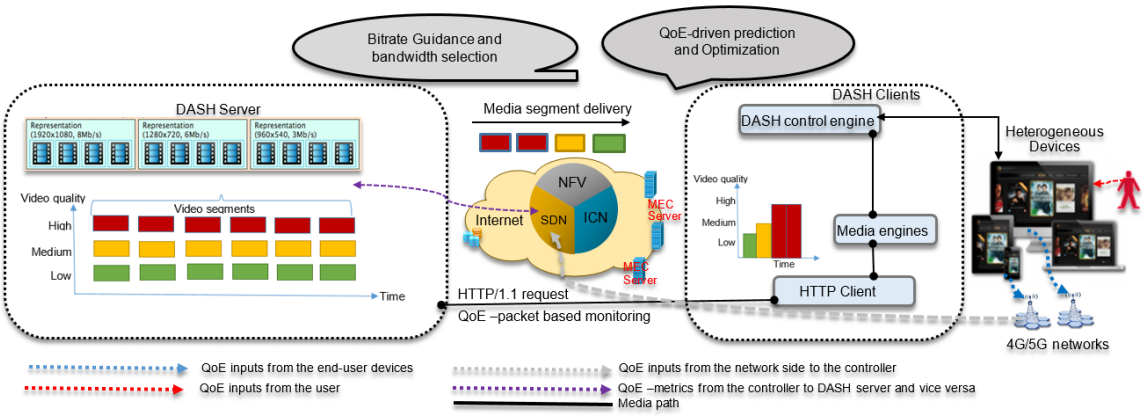

Figure 2.2: Dynamic adaptation according to network conditions

The client then measures/predicts the current bandwidth and buffer status and requests the next part of the video segment with an appropriate bit rate. This way, stalling (i.e., the interruption of playback due to empty playout buffers) could be avoided and the available bandwidth is best possibly utilized [56]. The success of HAS can be attributed to these benefits compared to the traditional streaming technologies: 1) video service providers can offer multiple quality levels by adapting bitrates and resolutions, and maybe other encoding parameters, of video to the end-users' demands; 2) different QoE-tailored personalized service levels and/or pricing schemes can be offered to customers and; 3) the demanded video quality based on the available bandwidth, can be adapted dynamically to changing network and server/CDN conditions. It is worth noting that DASH enables both live and on-demand delivery of media streams over HTTP. On-demand video streaming applications can benefit from optimized encoding methods such as multipass and variable bitrate encoding. As an example, Netflix has recently proposed a content-aware video encoding optimization scheme called the *per-title* encoding [57]. With per-title encoding, high-quality videos can be delivered even under low bandwidth conditions by streaming at higher resolution with the same bitrate. The basic concept behind changing resolution with bitrate is that below a given bitrate, it is better to encode a video at a lower resolution and then up-scale it at the end user device rather than encoding it at the higher resolution. Therefore, for the optimized encoding with the available codecs, using a per-title encoding scheme, each video is encoded into multiple resolution bitrate pairs, to obtain the optimum operating curve [57].

Despite success factors of a decentralized nature of HAS principle, there are still some drawbacks especially in the presence of multiple DASH clients that compete for the shared network resources. Some of HAS issues include (1) video instability due to constant bitrate

switching [58], (2) network resource under-utilization, and (3) QoE unfairness [3]. These problems remain a serious concern for video content providers and network operators, and they are even more aggravated in the case of heterogeneous environments. It is worth mentioning that using DASH, and it might therefore not be sufficient to optimize users' QoE from the available resources fully. Based on these reasons, we discuss next the MPEG-SAND standard where the centralized nodes within the network have been proposed recently to enhance the delivery of DASH content [2], [59].

2.3.2 Server and Network assisted DASH (SAND)

Server and Network assisted DASH (SAND) is an extension of the MPEG-DASH standard that has been recently published [1], [48] to enhance the delivery of DASH content. The SAND specification introduces messages between DASH client and network elements or between various network elements. SAND messages improve the streaming session by providing information about real-time operational characteristics of networks, servers, proxies, caches, CDNs as well as DASH client's performance [1]. SAND has been specifically designed to address: (1) content-awareness and QoE-service-awareness towards the underlying protocol stack including server/network assistance, (2) analytics and monitoring of DASH-based services, (3) unidirectional/bidirectional, point-to-point/multipoint communication with and without session (management) between servers/CDNs and DASH clients.

The SAND architecture shown in Fig. 2.3 consists of three categories of elements, namely, the DASH clients, DASH-Aware Network Elements (DANE), and regular network elements, which are DASH unaware and are present on the path between the origin server and DASH clients (e.g., transparent caches). DANE communicates with DASH clients while having minimum intelligence about DASH. For example, DANEs may be aware that the delivered objects are DASH-formatted objects such as the MPD or DASH segments. This way, they can prioritize, parse or even modify such objects. The MPEG-SAND standard reference architecture shown in Fig. 2.3 also defines three categories of SAND messages, namely: (1) the Parameters Enhancing Reception (PER) messages which are sent from DANEs to DASH clients for enhancing and improving their video quality adaptation, (2) Parameters Enhancing Delivery (PED) messages which are exchanged between DANEs, and (3) Metrics and Status messages which are sent from DASH clients to DANEs. DANE

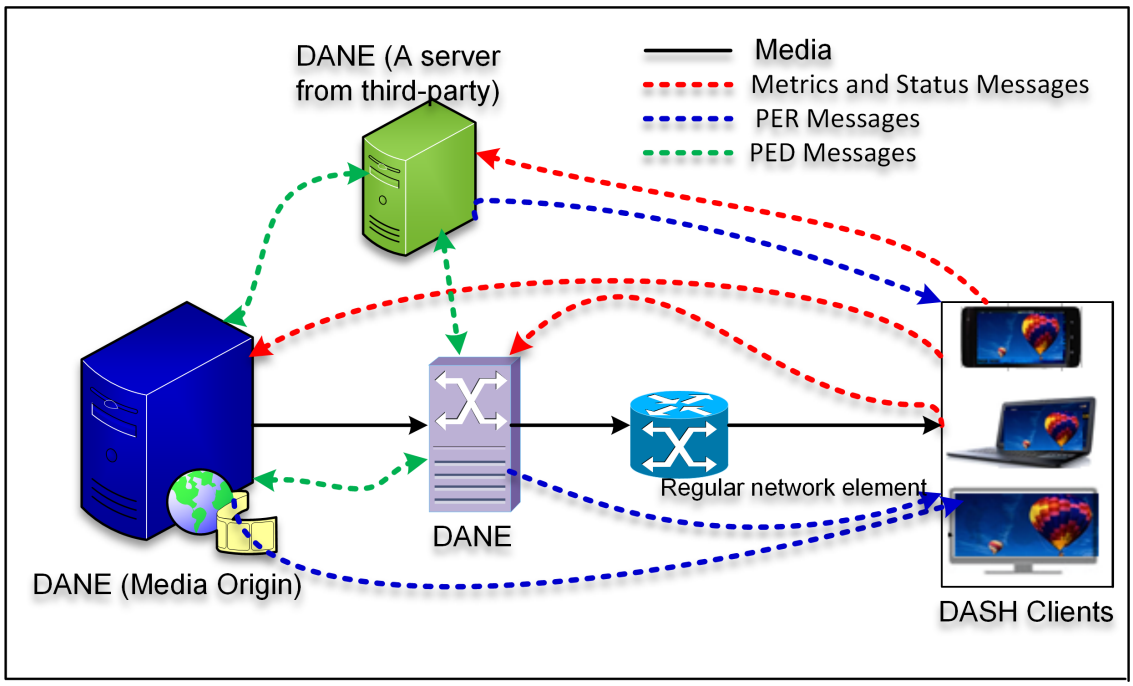

Figure 2.3: MPEG-SAND architecture. DANEs can communicate among each other and with the DASH clients to optimize the end-to-end video quality delivery [1]).

nodes become aware of the status of DASH clients using SAND status messages.

For example, QoE metrics reported by DASH clients to the network can be used for monitoring purposes and to simplify QoE-aware bandwidth optimization implementations. Desired quality and bandwidth requirement information reported by the DASH clients can facilitate resource sharing mechanisms among competing DASH clients. A DANE node can use PER messages to inform the DASH clients about the available network bandwidth. They can also inform clients about already cached segments by the DANE so that the clients can request these segments based on their device capabilities. Moreover, information about the streamed video to a particular network delivery element/node can be communicated by the server using a PED message. Generally speaking, a third-party server that receives messages concerning metrics from DASH clients and sends SAND messages to the clients is considered a DANE element.

It is important to note that the MPEG-SAND messages are delivered over HTTP using the Extensible Markup Language (XML) format and follow a specific syntax as defined by the standard in [1]. Compared to the client or server-based solutions, the MPEG-SAND standard represents an essential enabler for solutions that need collaboration between service/application and network providers. However, this needs a modification of network nodes/elements to provide a set of messages that can be exchanged by the

network, servers, and DASH clients during video quality delivery optimization [48]. For the practical and scalable implementation of the SAND approach, SDN can be used to provide a centralized control element. Efficient QoE-driven application and QoE-aware network management strategies using SDN and NFV appear to be vital solutions to guarantee the end user's QoE at a minimum cost.

2.3.3 Issues and Challenges of HTTP Adaptive Video Streaming

OTT multimedia streaming services are using HTTP adaptive video streaming these days where most of the services use MPEG-DASH standard as discussed in section 2.3.1. While the MPEG-DASH standard can provide excellent video quality to the end-users, there are some issues when multiple clients are competing for the shared limited resources leading to QoE unfairness, video instability and under-utilization of network resources [58], [3]. To overcome these challenges, the MPEG-SAND standard has been proposed recently with the aim to enhance the delivery of DASH contents, thanks to its ability to provide information of the network, servers, and streaming DASH clients in real-time. This way, it becomes easy for DASH clients to report information regarding video bitrates that can be downloaded from the server based on the available network bandwidth. Moreover, QoE measurements that are reported by the DASH clients can be used for monitoring purposes to perform QoE-aware bandwidth optimization efficiently. While SAND provides a set of well-defined messages between DANE and DASH clients during video streaming, the specifications leaves its implementations open in terms of physical network entities that are SAND capable. Some of the open questions regarding SAND implementations includes (a) What are the intelligent QoE-policies to divide network bandwidth among DASH players and other traffic over future softwarized networks, (b) How can network assisted DASH improve the streaming performance in terms of number of quality switches, video bitrate, number of freezes, and QoE-fairness over SDN/NFV-enabled networks, and (c) What is the effect of network bandwidth sharing/slicing policies on DASH Streaming performance over future SDN/NFV- enabled networks?

2.4 Network Softwarization and Virtualization: The Promise of SDN and NFV in Future Networks

Network softwarization [60] and virtualization using SDN and NFV are expected to impact several aspects of network development and services such as CDN or video accelerators [61], [62]. Before presenting the state-of-the-art solutions regarding QoE management using SDN/NFV in Section 2.5, an overview of the recent advancements in SDN and NFV as an important technology for management and orchestration of resources in future networks is discussed in section 2.4. In that respect, section 2.4.1 and 2.4.3 provide background information of SDN and NFV respectively. This thesis uses network softwarization and virtualization concepts presented in this section in QoE management aspects implementations.

2.4.1 Software Defined Networking (SDN)

SDN Definition

SDN [14] is an approach that brings intelligence and flexible programmable networks capable of orchestrating and controlling applications/services in more fine-grained and network-wide manner [63]. The Open Network Foundation (ONF) [64] defines SDN as "*the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices*". Notable advantages of the SDN architecture include enhanced network programmability, centralized control, and management, increased network flexibility and reliability, data flow optimization [64].

SDN Architecture Design

SDN creates a virtualized control plane that can enforce intelligent management decisions among network functions bridging the gap between service provisioning and QoE management. The main layers of SDN architecture and its design components are illustrated in Fig. 2.5. The SDN architecture consists of three layers, namely: the infrastructure, control and application layers.

- **Control Layer:**

The control layer is the brain of SDN which consists of the component called the "controller". The SDN controller is responsible to forward and process policies for the network flows. The control layer of an SDN network can be implemented as a pure software that runs on commodity hardware devices. The network service functions illustrated in Fig. 2.5 will be implemented in the Control Plane as demonstrated from Chapter 3 to 6 in this thesis. With SDN, the network control becomes directly programmable using standardized Southbound Interfaces (SBI) such as OpFlex [65], FoRCES [66], and OpenFlow [67]. The forwarding plane of SDN can be implemented on a specific commodity server [68] such as VMware's NSX platform [69] which consists of a controller and a virtual switch (vSwitch).

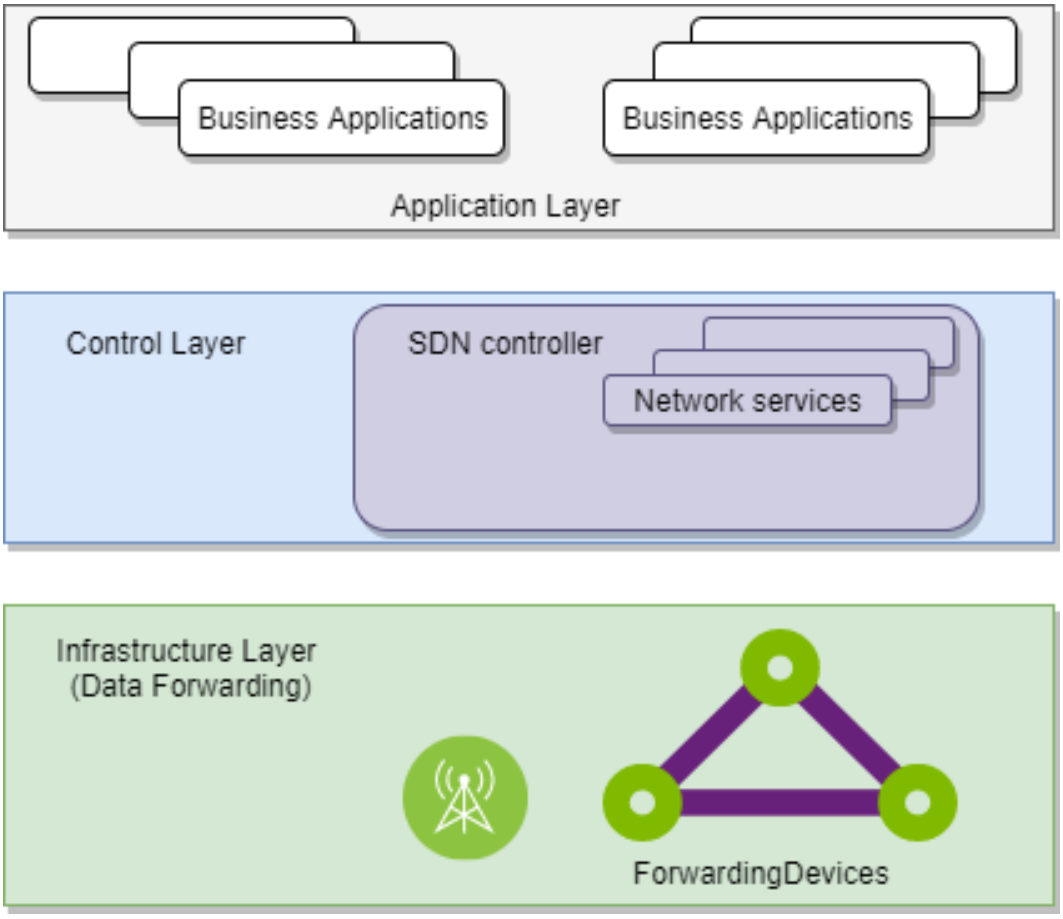

Figure 2.4: Main layers of the SDN architecture.

The design and implementation choice of SDN controller significantly affect the overall performance of SDN-based networks [14]. Key design and architectural choices for the controller are: *centralized* and *distributed*. A centralized SDN controller is best suited to manage small networks or a single domain. However, network management using the centralized controller can be confronted with scalability and reliability problems due to

the incremental loads on the controller [70]. The controller itself can be a single point of failure which can negatively affect the performance of the network and the end-users' QoE. SDN controllers such as Floodlight [71], Beacon [72], Ryu NOS [73] and POX [74] have been designed to achieve system throughput in specific environments such as data centers, carrier-grade networks, and cloud infrastructures. It is vividly important to note that a centralized controller is not enough for large-scale network management. As such, scalability and reliability in large-scale networks that span multiple control domains can be achieved by distributed SDN control architectures such as Hyperflow [75], HP VAN SDN [76], DISCO [77], and ONOS [78]. In essence, a distributed SDN controller implementation can be either a set of elements distributed physically or a centralized cluster of nodes [14]. It is worth mentioning that the distributed SDN controller reduces the network partitioning[2] problems and improves the scalability and resilience of the control plane in SDN [79].

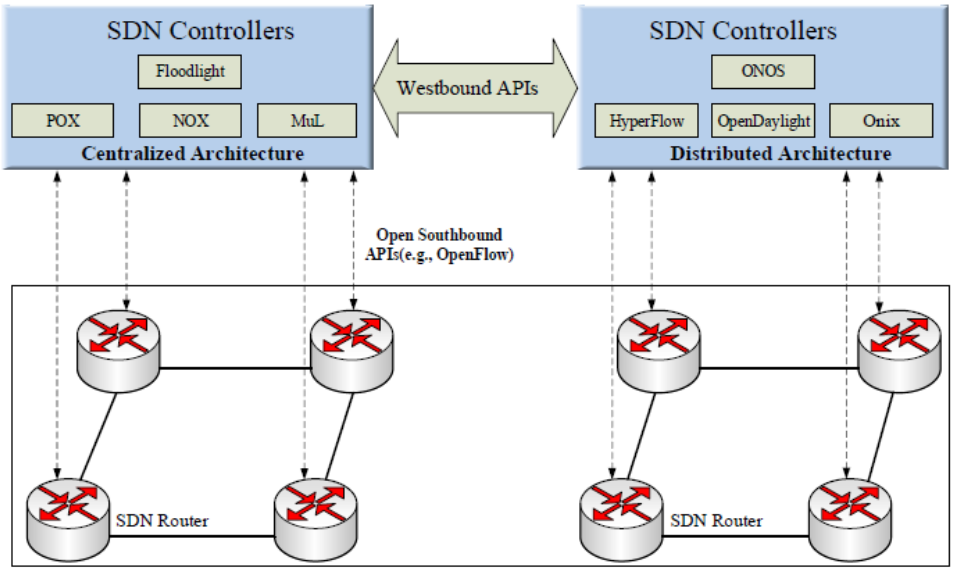

Figure 2.5: Illustration of information exchange between SDN controllers using Westbound APIs

- **Application Layer:**

The application layer is where SDN-enabled applications are created and defined to run on top of the SDN controller. The defined applications are responsible to manage the network by making decisions based on changing network conditions (e.g., during link/node failure). Some of the SDN applications scope include the following: load balancing,

[2] A network partitioning refers to network decomposition into relatively independent subnets for their separate optimization as well as network split due to the failure of network devices

network monitoring and management, QoS- enforcement policy, security and access policies, traffic engineering etc. The Northbound API enables communication between the SDN application and the control layer [14]. Although there are no standardized interface, most of SDN controllers today have RESTful APIs used to communicate between the SDN Controller and the services and applications running over the network [14]. It is important to mention that, SDN applications can be divided into two groups namely, (a) SDN-based internal applications (b) SDN-based external applications. The former is is implemented to run inside the SDN controller while the latter is deployed and runs outside the SDN controller or outside the container where the controller is hosted.

- **Infrastructure Layer:**

This layer represents the data-forwarding plane where both software/hardware devices (e.g., OpenFlow Switches, routers, middleboxes, virtual switches, firewalls) are interconnected via wired or a wireless radio channels. One or more data paths form a route for forwarding network traffic in the network based on their flow tables. The SDN controller defines the routing rules of flow tables. A set of packet field match values (source and destination MAC addresses, source and destination IP, VLAN, port number etc) and a set of actions (e.g., delete, modify, forward to port) are used to represent each entry in the flow table. An OpenFlow protocol is used to control and manage communications between the infrastructure and control layer. The data-forwarding plane can collects the raw measurement data from the network using passive or virtual probes [35].

2.4.2 OpenFlow Standard

OpenFlow protocol [67]is an open standard that defines communication between data forwarding plane and the SDN controller layer. This allows to decouple the data plane and control plane while also enabling higher programmability in SDN. Today, OpenFlow is supported by different networking and service provider's company to implement SDN applications related to QoS/QoE monitoring and management [80], [26]. An OpenFlow switch stores forwarding rules that are sent from the SDN controller. The OpenFlow switch takes this forwarding decision based on the port number where this packet is coming from. It is important to mention that, the SDN controller takes control of the forwarding rules and their computation in SDN: this enable switches to have minimum requirements in terms of memory and CPU for processing data in SDN. Figure 2.6 shows

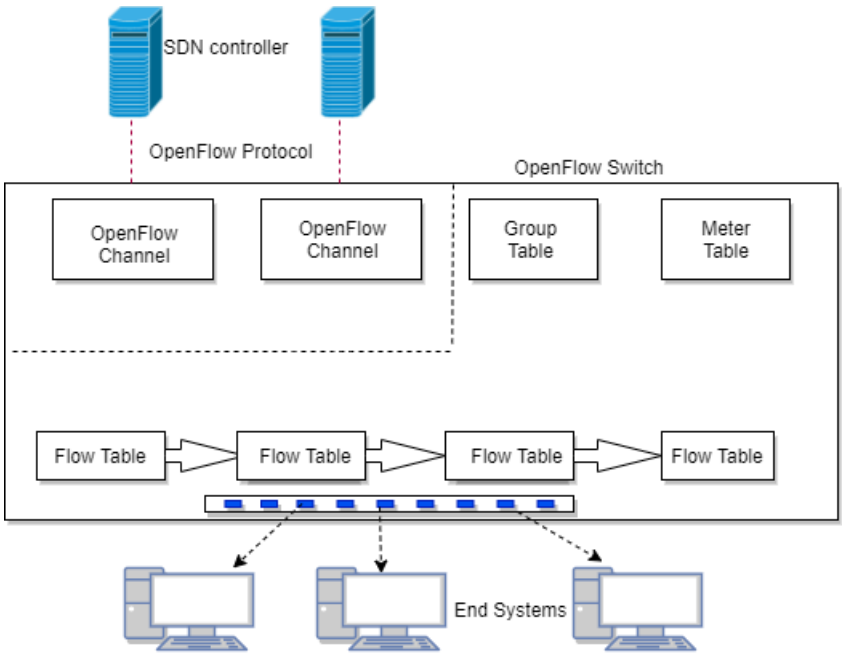

Figure 2.6: An OpenFlow architecture

an OpenFlow architecture consisting of a Flow Table, Group Table and Meter Table.

These tables are mainly used for performing table lookups and forwarding decisions. An OpenFlow switch consists of one or more OpenFlow channels which connect directly to an external SDN controller. The controller is responsible to manage the switches through the OpenFlow protocol. The flow table represents the packet forwarding table in OpenFlow switch. A flow table consists of flow entries. Each flow table entry contains:match fields, priority, counters, instructions and timeouts cookie. A group table consists of group entries which allows an OpenFlow to offer an additional forwarding methods.

Figure 2.7 indicates the meter table defined in OpenFlow v1.3. The main components of a meter entry in the meter table are (a) The meter table offers the QoS mechanisms in SDN. It achieves this by measuring the flow rate of data packets. It also keep monitoring the rate of data packets prior to output. The flow entry in the table is called a meter. Each meter is attached directly to a flow entry in the table of an OpenFlow switch. The rate of network traffic which are defined by a specific flow can be measured, controlled and monitored in real-time by a meter. By using the goto-meter action, data packets in SDN can be directed in flow tableâĂŹs entry to a meter. The meter can then perform different operations on the data a packet such as to limit the flow rate or combine with QoS queues in order to implement different QoS mechanisms for controlling and managing multimedia services.

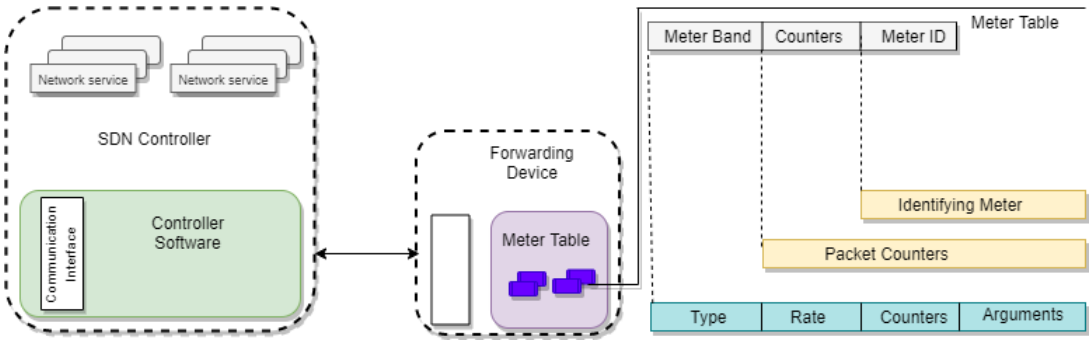

Figure 2.7: Meter Table defined in OpenFlow v1.3

2.4.3 Network Function Virtualization (NFV)

NFV [15] is the decoupling of physical network equipment from the network functions such as Firewalls, Deep Packet Inspection (DPI) that run on them. NFV envisages the instantiation of Virtual Network Functions (VNFs) on commodity hardware. This way, it breaks the unified approach to functional software and hardware that exists in traditional vendor offerings. With NFV, Network Functions (NFs) can be quickly deployed and dynamically allocated. Also, network resources can be efficiently allocated for these VNFs through dynamic scaling to achieve Service Function Chaining (SFC)[3]. For ISPs, NFV promises to provide the needed flexibility that would enable them to support new network services faster and cheaper to realize better service agility and to reduce their Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) through lower-cost flexible network infrastructures. NFV also aims to decrease the deployment time of new network services to market through innovation cycle of software-based services deployment. To achieve the above benefits, NFV brings three differences in the network services provisioning as compared to traditional practice [82, 15] by (a) decoupling software from hardware platform, (b) providing greater flexibility for NFs deployment, and (c) enabling dynamic network operation and service provisioning. It is important to note that while the full-blown software-based implementation using SDN and NFV concepts come with these benefits, the question is whether the design considerations can meet some technical performance requirements needed by Telco Cloud or service providers. A blueprint of NFV management and orchestration framework as proposed by the European Telecommunication Standard Institute (ETSI) is discussed next.

The NFV concept in operator infrastructures [83] was first explored by the ETSI, mostly

[3]Service Function Chaining (SFC) is an ordered list of general service functions that should be applied to a packet and flows selected as a result of classification [81].

to address the challenges towards flexible and agile services and to create a platform for future network monetization. Since then, the NFV reference architecture shown in Fig. 2.8 was proposed [84] followed by a proof of concept (PoC) [85]. The ETSI Management and Orchestration (MANO) framework consists of functional blocks which can be grouped into the following categories: the NFV Infrastructure (NFVI), NFV Management and Orchestration, Network Management System (NMS) and VNFs and Services. These entities or blocks are connected using reference points[4].

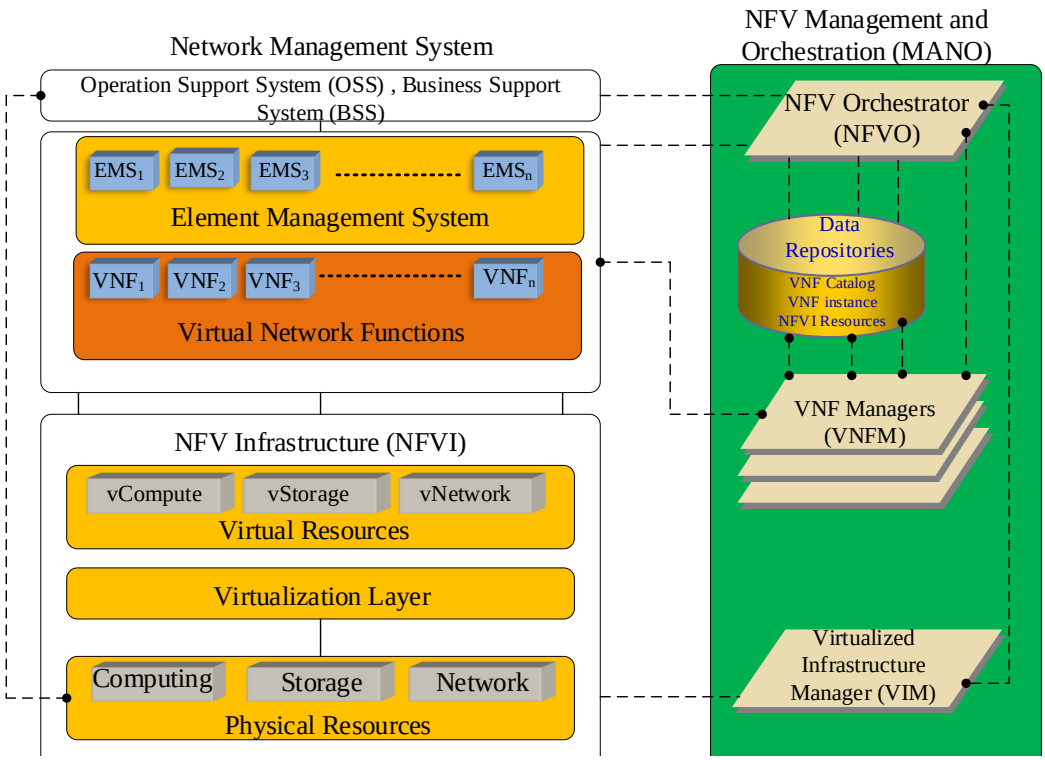

Figure 2.8: The ETSI NFV management and orchestration framework.

The NFVI forms an environment consisting of both physical and software/virtual resources. Physical resources indicate the computing hardware (e.g., for processing), storage and network resources. Virtual resources are the abstractions of the computing, storage and network resources achieved through a virtualization layer, for example, based on hypervisor which is a typical solution for VNFs deployment today. The Network Management System (NMS) is a critical aspect of the overall operation of both VNFs and physical resources. It potentially deals with functions related to network management such as fault management, security, configuration and performance management. The fault management provides a key role for QoE assurance by making sure that network failures or problems are recovered before users are disconnected from the network services.

[4] A reference point defines a point where two communicating functional entities or blocks are connected

According to the ETSI [82], the NFV Management and Orchestration (MANO) is responsible for managing and orchestrating VNFs. It consists of the NFVO, VNFM and the VIM. The NFVO performs orchestration and lifecycle management of physical and software resources that support the virtualized infrastructure. The NFVO also performs global resource management, network service instantiation, validation, and authorization of NFVI resources requests. The VNFM is responsible for lifecycle management of VNF instances, overall coordination and adaptation role for configuration and event reporting between NFVI. A single or multiple VNF instances of the same or different types can be managed by a VNFM. The VIM controls and manages NFVI physical and virtual resources (vCompute, vStorage, and vNetwork resources) within one operator's infrastructure domain. The NFV MANO consists of a database that keeps the VNF catalog, VNF instances, NFVI resources, and network services catalog. The VNF instances keep information of all VNF instances and network service instances while the NFVI resources repository holds information of all allocated or available NFVI resources.

Table 2.2: The relationship and comparison between SDN and NFV

Category	NFV (Telecom Networks)	SDN (Data Center Networks)	Already Adopted
Network Control	Seamless control and dynamic provisioning of NFs	Provide a centralized network control	Yes
Architectural Design	Service or NFs abstractions	Networking Abstractions	Yes
Main Advantage	Offering flexibility needed by network	Offering programmable network with open control interfaces	Yes
Cost Efficiency	Replace hardware with software	Operational efficiency and energy consumption reduction	Yes
Standard Protocol	Supporting multiple control protocols	OpenFlow is the de-factor standard protocol	Yes
Leaders/Business Initiator	Born in Telcom Service Providers	Born for networking software and hardware vendors	N/A
Formalization	ETSI	ONF	N/A

2.4.4 Network Hypervisors, Containers and Virtual Machines

Network hypervisors [86] are the network elements that abstract the physical infrastructure (e.g., communication links, network elements, and control functions) into logically isolated virtual network slices. In physical SDN network, a network hypervisor offers high-level abstractions and APIs that greatly simplify the task of creating complex network services. Moreover, the network hypervisor is capable of inter-networking various SDN providers together under a single interface/abstraction so that applications can establish E2E flows without the need to see or deal with the differences between SDN providers [87]. Through hypervisors, it is possible even to implement higher layer services such as load balancing servers and firewalls or link and network protocol services belonging to L2 and L3 [15].

Virtual Machine (VM) [88] enables the virtualization of a physical resource where an

experimenter can run his/her own Operating System (OS). The basic principle of a VM is that resources such as computing, storage, memory, and network are shared among VMs. However, the entire operational functions of a VM is isolated completely from that of the host and another guest VMs [89], [90]. It is also possible to run multiple VMs at the time on one physical machine.

Containers are light-weight alternatives to hypervisor-based VMs [91] and are created based on the idea of OS-level virtualization. A physical server in containers is virtualized such that standalone applications and services can be instantiated on an isolated servers [89]. Different from VM-based counterparts, containers do not need hardware indirection and run more efficiently on host OS leading to higher application density. Examples of container-based virtualization include: Docker [92], Linux-Vserver [93], OpenVZ [94], and Oracle Solaris Container[5]. In this vein, VMs and containers are capable of running VNFs chained together to deliver a 5G network service or application flexible and therefore forming a base functionality for 5G network slicing. It is important to note that, while containers can efficiently support 5G network slices with highly mobile users, VMs may offer full logical isolation for operating VNFs in a network slice [89].

2.5 QoE Management for HTTP Adaptive Video Streaming using SDN/NFV: State-of-the-Art

Since the MPEG-DASH standard was defined, researches from both the academia and industry have investigated how to improve the end-users' QoE [95]. Many studies have been focusing on new heuristic solutions to improve the QoE based on video quality adaptation on the client or server-side actions [11, 96]. In recent years, the attention has been towards developing approaches that fully optimize the video quality [97, 2] and achieve users' QoE-fairness level [98]. As of today, many SDN/NFV - based adaptive solutions have been proposed, implemented and tested to see the value and benefits offered by these technologies, recent efforts include, a QoE-aware bandwidth broker [99] and a rate-guided QoE-aware SDN-APP [100]. This section present different QoE-management strategies using SDN/NFV. The work focus mainly on three research areas including server and network-assisted optimization approaches in section 2.5.1 followed by QoE-Fairness and Personalized QoE-centric control in section 2.5.2. Section 2.5.3 provides a detailed QoE-

[5]https://www.oracle.com/technetwork/server-storage/solaris/containers-169727.html

centric routing while section 6.2 presents QoE-softwarization concept that was proposed by the author in [63]. This thesis employs and builds on relevant works described in this section to design and implement novel algorithms for managing multimedia streaming services in future networks.

2.5.1 Server and Network-Assisted Optimization Approaches using SDN

Despite the success of client-based adaptation approach for HAS streaming presented in subsection 2.3.1, it may not be sufficient to provide fair distributions of bandwidth between users. Thanks to the recent development of SAND where the availability of a controller that has a view of the network nodes status and can control all their functions become an optimal match of its objective [95].

Following the development of SAND, bandwidth shaping and bitrate guidance strategies have attracted much attention in the multimedia streaming community [12], [101], [2], [99]. Most research works following this approach share a similar architecture as shown in Fig 2.9. Key quality parameters from both the network and the client side can be collected using a centralized node. The client-based information collected may include, for example, the device type [12], the bitrate of the video [102], buffer level [101] and the screen dimension. The network QoE-related parameter measurements include the number of streaming DASH clients and the available bandwidth. That way, the centralized node in an SDN-assisted environment has a comprehensive view of the streaming multimedia service. It is hence possible to select the best bitrate for each client to optimize an objective function that models the users' QoE. For the practical and scalable implementation of this approach, SDN can be used to provide a centralized control element.

Authors in [2] compare the performance of bitrate guidance and bandwidth shaping techniques. The optimal bitrates for DASH clients are computed by the SDN controller to achieve video quality fairness which is calculated based on the SSIM index. When *Bandwidth Reservation* is used, the bandwidth slice is assigned to clients with similar optimal bitrates. As shown in Fig 2.9 (a), two control loops, *inner and outer control* loops are used. Based on the video client feedback and bandwidth estimates, the inner control loop running at the client side selects the video bitrate. The outer control loop is executed in the network and sets the bandwidth slice. In the bitrate guidance scenario shown in Fig 2.9 (b), the optimal bitrates are computed by a centralized algorithm running in a network

element. The video bitrates are then communicated to the DASH client that download the corresponding video segment. It is imperative to mention that the bitrate guidance provides the best performance regarding switching frequency and video quality fairness compared to the bandwidth shaping approach [95].

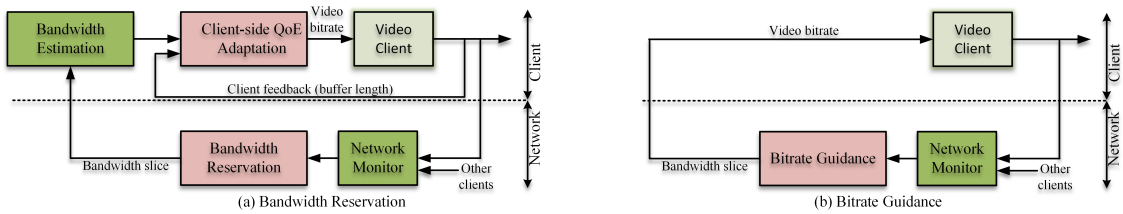

Figure 2.9: QoE-based network-assisted approaches for adaptive video streaming. (a) A specific bandwidth for each client is enforced by the network when bandwidth shaping is used, and (b) the network provides an explicit bitrate to the clients using bitrate guidance [2].

Following the same approach, authors in [103] propose a network-level QoS mechanism to enforce application QoE-aware fair resource sharing among competing clients. Using the QoE model that depends on screen size and resolution, the network controller selects the best video bitrate to maximize the QoE for each client. A Video Home Shaper (VHS) is designed to monitor outbound HTTP requests and capturing those that identify Netflix and YouTube sessions initiated by clients connected to the router. The QoE-fair bandwidth for all active streaming sessions is computed and recorded by the session manager. The bandwidth manager then allocates lower and upper bandwidth value to each video stream. The lower value is the guaranteed minimum bandwidth allocated for that session while the upper value indicates the maximum allowable bandwidth.

An SDN-assisted Adaptive Bitrate Streaming (SABR) is proposed in [59]. The available bandwidth per link and network cache contents are the information used by SABR to guide the QoE maximization of the clients. SABR provides the DASH client with monitoring information (bandwidth estimates and cache occupancy) through a REST API. SABR use dynamic SDN routing to provide clients with the ability to connect to the desired cache. It is worthwhile emphasizing that SABR not only significantly improve the QoE (e.g., the overall video quality bitrate) at the client side but also reduces the server load ratio and provides higher network utilization.

A network-QoE-aware video segment selection and caching approach, *"QoE-SDN APP"* in the context of HAS is proposed recently by [100]. Using QoE-SDN APP, video service providers (VSPs) can program the end user's QoE requirements. The core logic of the

QoE-SDN APP is the QoE Assessment module which performs the following tasks (a) determine and recommend the encoding rate and caching strategy for VSP while considering the future network load and user mobility, and (b) determine the QoE per application using application-specific KPIs such as MOS scale. VSP-QoE Control Agent is another module within the SDN controller that provides feedback to the VSP regarding the control capabilities related to the data plane and allows VSPs to collaborate with the underlying MNO's infrastructure. That way, VSPs can quickly enhance their distribution procedures and video segment encoding by using network feedback exposed by the MNOs.

[99] proposes a QoE-aware bandwidth management solution for HAS flows in an SDN-enabled Hybrid Fiber Coax (HFC) access networks. To improve the end user's QoE, important parameters considered include video content resolution, encoding bitrate, video type, device capabilities, and service plan types. Such parameters are used by the bandwidth management application to enforce QoE control and management strategies such as bandwidth allocation and provisioning, network resource management, representation decision, reservation and monitoring, and QoE-tailored service/flow prioritization and differentiation [99].

2.5.2 QoE-Fairness and Personalized QoE-centric control in SDN

We provide in this subsection a detailed discussion on QoE-fairness and personalized QoE-centric control approaches for adaptive video streaming. Most of the works presented in this subsection utilize the SDN controller to manage and monitor all HAS players, their statuses, device capabilities, requested content, subscription plan types, QoE and buffered level. In this way, the controller can easily detect player-specific events, such as players joining/leaving the network and starting/stopping the playback. Some of the approaches for example in [12, 101] employ an optimization function that interacts with the controller to dynamically optimize QoE fairness of multiple competing clients by setting the bitrate for each streaming video in the network. Based on the principles of SDN and Participatory Networking (PANE) [104], [105] proposes a client-Driven Video Delivery (cDVD), a proof-of-concept that provides a client-level API into the network. cDVD relies on a client-driven network-level QoE fairness approach that provides stable video quality by maintaining low re-buffering ratios in an encrypted SDN-assisted environment. cDVD at a high level is set to enable one or more DASH clients to interact with network components

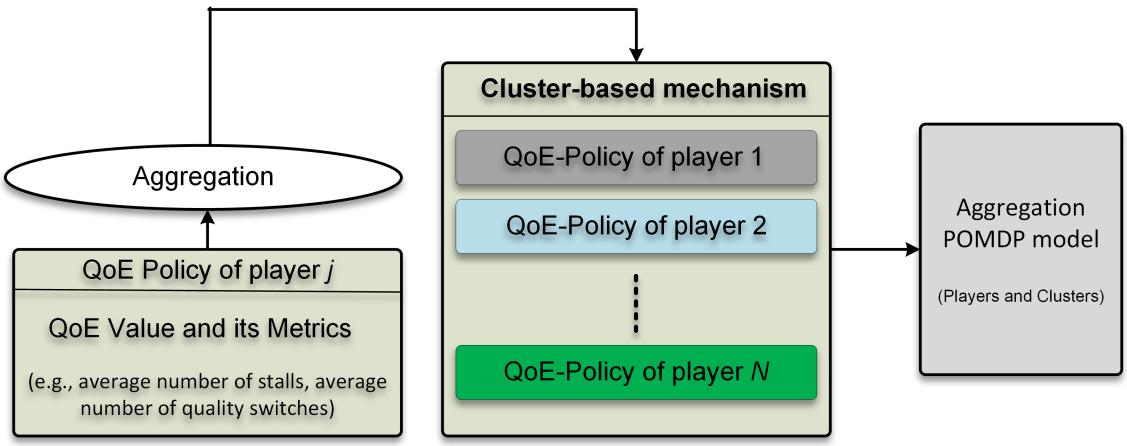

Figure 2.10: Per-cluster QoE policy structure and model abstraction (Adapted from [3]).

(e.g., SDN controller) to achieve session-level QoE-fairness. As presented in the cDVD architecture [105], each session can comprise of multiple flows where the controller applies bandwidth control to the routers using a suitable mechanism, such as OpenFlow or direct queue configuration. The controller uses the bandwidth enforcer to limit the rate of each flow and in turn allocates bandwidth to equalize QoE across sessions. This is achieved by applying a QoE-based Utility module to determine the fairness of video streams [103].

An OpenFlow-Assisted QoE Fairness Framework (QFF) is introduced by [12] to provide the required user level QoE in multimedia networks. QFF can allocate network resources and ensure that the maximum number of users with the target QoE fairness level is achieved in a heterogeneous environment. In QFF, OpenFlow protocol allows vendor-agnostic functionality to be implemented for network management and active resource allocation. As such, the status of the network and DASH video streaming sessions are monitored by QFF using the Network Inspector and the MPD Parser module respectively. The QFF in-turn dynamically allocates network resources to each client to equitably maximize users' QoE in multimedia networks. The network intelligence of QFF is provided by the Utility Function (UF) and the Optimization Function (OF). The UF provides a model that maps the bitrate of a video to the QoE delivered on a particular client's device. For each video streaming session in the network, the OF is responsible for finds a set of video bitrates that will provide a QoE-fairness level for all DASH clients [12].

SDNDASH, a dynamic network resource allocation and management architecture for HAS systems is proposed by [101]. SDNDASH avoids quality instability, unfair bandwidth sharing and network resource underutilization among competing DASH clients sharing the same bottleneck network link. In this way, the per-client QoE is optimized while

reaching the required maximum QoE-fairness level. As an extension to this work, Bentaleb *et.al* [3] propose an SDN-based streaming solution called SDNHAS that can assist HAS players in making better QoE adaptation decisions. SDNHAS can optimally implement the target QoE policies for a group of users and allocate the network resources efficiently in the presence of both short and long-term changes in the network. One of the central SDNHAS entity is an optimizer component shown in Fig. 2.10 that forms a logical network topology to group the HAS players into a set of virtual clusters. A specific data structure for each cluster, called the *per-cluster QoE policy*, is constructed during each video segment being downloaded. To make a fair allocation of bandwidth, each player has its own QoE policy that also includes QoE values and its metrics. A set of players, whose QoE policies belong to the same cluster are aggregated together into a common per-cluster QoE policy using a simple aggregation [3].

To solve video representation selection problem, a Partially Observable Markov Decision Process (POMDP) optimization model is used that takes different input variables such as congestion level, device resolution, content type, buffer level, subscription plan type, and the available bandwidth. POMDP applies to multi-agent decision problems and considers observation and historical information capabilities (of state definitions), unlike other mathematical models (e.g., Markov Decision Process model (MPDM), which only supports single agents [3]. With POMDP, it is easy to get partial information on the system state. It is, therefore, the best approach in the network environment with dynamics such as HAS systems that exhibit high uncertainty like sudden bandwidth fluctuation. It is important to note that SDNHAS, provides QoE-aware adaptive streaming delivery and intelligent network management that enables a maximum level of user satisfaction among heterogeneous HAS players.

Following the same approach of QoE-fairness and QoE-personalized control, an SDN-based multi-client bandwidth management architecture for HTTP adaptive video streaming that can support up to 75% users at the same QoE level is proposed in [106], while a Q-learning-based dynamic bandwidth allocation strategy to achieve QoE fairness is given in [107]. A user-level fairness model, UFair, which orchestrates network resource allocation between HAS streams to mitigate QoE fluctuations and improve the overall QoE fairness is given in [98]. UFair uses the video quality, switching impact and cost efficiency metrics to measure the user's QoE fairness. Following a specific adaptation and fairness criteria of equal bandwidth for every active DASH client, high-quality videos are

delivered over a DASH-aware SDN-based architecture in [102]. While heading towards 5G networks, what is interesting and remains to be seen is how QoE-fairness [108] metrics can be integrated as the benchmark for QoE management and optimization in future networks.

2.5.3 QoE-Centric Routing Mechanisms using SDN/NFV

Efficient delivery of video streams with improved QoE can be achieved in SDN using shortest paths, multiple disjointed paths or IP multicast procedures. For example, using the concept of Economic Traffic Management (ETM)[6], a QoE-centric routing approach that utilizes QoE estimation models to maximize the user QoE for multimedia services is presented in [109]. The QoE measurement and collection of QoE Influence factors (IF) values in an SDN network is achieved through the cooperation among various components including the end user's clients and media content servers and SDN controllers. The clients and content server report the QoE values to the SDN applications which in-turn use the reported QoE values and IFs to prepare the needed input for the SDN controller. Subject to traffic demands and network constraints, the controller can select a path that can maximize QoE for DASH clients.

A QoE-driven path optimization model (Q-POINT) that maximizes the end users' QoE through the best path calculations for each service flow is given in [110]. In order to negotiate important parameters for video streaming sessions that are to be established, Q-point utilizes the Session Initiation Protocol (SIP)[111] which is assisted by the SIP application server with QoS-QoE mapping functions (for different media types) and Optimization Function (QMOF). For each session, the QMOF calculates a set of configurations that include information such as media codecs, video bit rates, and user preferences. In turn, the SDN controller determines which multimedia flows are to be routed along a particular path in order to maximize the aggregated QoE. The routing decision by the SDN controller is made based on the calculated session and media flow parameters.

A QoE-driven multimedia service optimization and path assignment architecture are proposed in [112], with the main idea of including various application-level network functions involved in the negotiation and QoE-optimization decision making as highlighted in [113]. A software-defined scalable multimedia multicast streaming (SDM^2Cast) [114]

[6]ETM is an economic incentive that enables a Triple-Win solution for users, service providers and network operators. It aims to reduce cost in the network while providing QoE-enriched services to the end-users

is introduced to provide in-network adaptation by adjusting the number of layers, which in turn improves the end-users' QoE. Based on the current network state, SDM^{2}Cast can flexibly customize network layers multicast paths and thereby allowing SDN implementations to recognize, process and manage media streams through in-network adaptation. An SDN-based video streaming multicast application that performs the calculation of routes and multicast trees after mapping clients to servers with the highest QoE is presented in [115].

Athanasopoulos *et.al* [116] propose a quality-aware path switching scheme, along with a media-aware optimization algorithm that selectively shapes the video traffic according to the access network conditions. Nam *et.al* [117] introduce a QoE-aware SDN based video streaming framework that dynamically changes routing paths using Multi-Protocol Label Switching (MPLS) Traffic Engineering (TE) to provide to clients a reliable video watching experience. A Scalable QoE-aware Path Selection (SQAPE) scheme for large-scale SDN-based mobile networks is demonstrated by [118]. Using a centralized control strategy of SDN, SQAPE provides fine-grained control for per-user QoE-aware path deployments across the network. As shown in Fig. 2.11, SQAPE consists of three main components (*Decision, QoS Measurement, and QoE Predictor*) which are decoupled from the SDN controller.

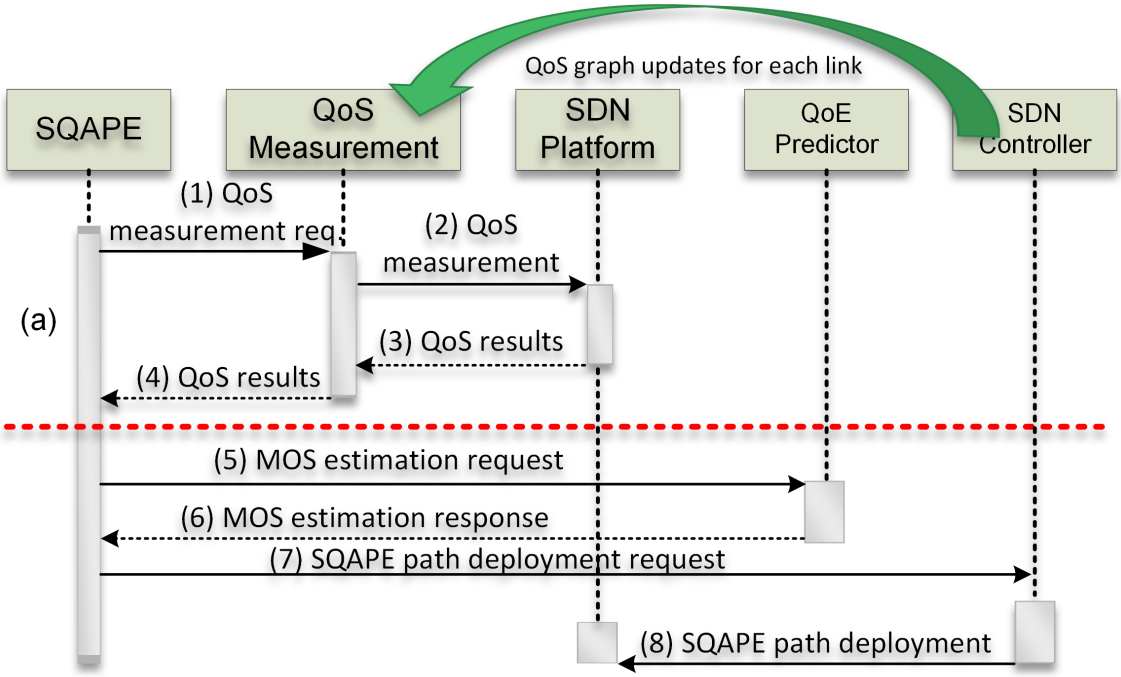

Figure 2.11: Scalable QoE-aware path selection in large-scale SDN-based networks.

Steps (1 - 4) involve QoS measurements to predict the QoE and the performance of video

streaming. The periodic QoS monitoring frequency of 60 seconds is used to perform active measurements based on packet loss, delay, and bandwidth of the link. Such metrics are selected based on previous investigations of QoS to QoE mapping which found these metrics as the most influential to video streaming QoE [119]. SQAPE in step (5 - 8) relies on the QoS-to-QoE mapping function to compute per-path MOS estimation. The estimated MOS along with a link utilization heuristic is then used to compose the distance metric which in turn, determines optimized QoE-aware paths to be deployed in SDN-enabled networks. An optimal path for each client is recomputed by [120] each time a new video segment is requested.

The video bitrate requested by a client is sent by a server to the SDN controller. The controller uses this information and the links status to reroute the traffic to maximize the throughput of the overall link. QoE-aware routing can also be performed in the existence of multiple servers where the client selects the best server to stream the video from in a dynamic way [121] [122]. [123] proposes a learning-based approach that minimizes the packet loss rate, quality changes, and controller cost while adapting the flow routes and video quality. Gangwal *et.al* [124] propose an Efficient Layer Based Routing Algorithm (ELBA) that exploits the dynamic re-routing capability of SDN, to different stream layers of Scalable Video Codec (SVC) video over distinct paths. ELBA improves network resources utilization and optimizes the video delivery, based on the routing decision of the SDN controller that assigns distinct routes for each video layer. It is important to note that traffic routing improves QoE by continuously monitoring the performance of the paths that connect the server and clients. However, novel paradigms such as MPTCP and SR can facilitate an efficient routing and speed up the transfer of large amounts of multimedia applications through multiple disjointed shortest paths.

2.6 Summary

QoE concepts are more user-centric than QoS which made it a topic of high interest in the multimedia communication research community for the QoE-based user-centric service management of the streaming services. The QoE-aware management of the multimedia streaming services consists of the three major components: *(1) QoE Modeling and Assessment* that includes subjective assessments by the real users and creation of the predictive mathematical models (QoE model) from the data related to KQIs that is acquired after

subjective assessments; *(2) QoE Monitoring and Measurement*– that includes the monitoring of the KQIs of the multimedia streaming service and measurement of the QoE from the predictive QoE models; *(3) QoE Optimization and Control*– mainly to perform optimization of the resources (network resources, content delivery network or client side adaptation) through control actions based on the QoE measurements. DASH has been adopted as an adaptive bitrate streaming technique to enable high quality streaming of media contents over the Internet delivered from conventional HTTP web servers. However, some of the drawbacks of DASH include QoE unfairness, video instability and under-utilization of network resources [3], [58] when multiple clients are competing for the shared limited resources. SAND offers standardized interfaces for service providers and operators to enhance streaming experience by using message exchanges between DASH clients and DANE. However some of the serious questions regarding SAND is how network assisted DASH can improve the streaming performance in terms of number of quality switches, video bitrate, number of freezes, and QoE-fairness. From the above literature review, network softwarization and virtualization paradigms are foreseen to be an important elements for QoE management of future multimedia services.

Chapter 3

Video Quality Management based on Queueing Mechanisms and QoE/QoS Policy in SDN

Queuing mechanisms can guarantee the QoS/QoE of traffic with higher priority flow when there are multiple types of flows sharing the network path from source to destination. Using the defined QoS/QoE policies and the configured queue mechanisms, traffic flows can be specifically classified for only users who demand high QoE requirements from service providers than other users. For example, the defined QoE flow polices for premium users can be forwarded to the higher priority queue to acquire resources that demand sufficient bandwidth while standard users' traffic flows can be forwarded to the lower priority queue. The SDN controller can differentiate the type of flows such as multimedia flows based on QoS/QoE requirements and allocate required resources to these flows. In the context of SDN virtualized networks, this can be done through mapping of traffic flow to virtual networks, composed by queues of appropriate links which meet its performance and QoE requirements (e.g., premium or standard users).

3.1 Introduction

SDN has the ability to provide secure and reliable end-to end communications that can satisfy specific transmission requirements as demonstrated by Qin *et.al* [125]. This enables SDN to achieve stringent requirements on network transmission latency for specific networking services and bandwidth intensive applications such as video streaming and online gaming.

All transmissions in the forwarding plane of SDN involve the controller where forwarding

devices (e.g., switch or router) has to request and apply forwarding rules when the first packet of each video flow arrives at the forwarding device. Frequent communications also involve the controller whenever there is an update of forwarding rules or statistical data requests by the controller to the forwarding device. In this context, the SDN controller becomes the bottleneck in the SDN architecture and thus, can cause QoE degradation of delivered multimedia services. It is therefore, crucial to understand the performance of the SDN architecture under difference traffic patterns, especially the bursty nature of video streaming. SDN can issue instructions and forwarding rules in order to modify network flows when needed and take a different path if network congestion is detected in one of the forwarding devices due to queuing delay. Modifying network flows by switching to low congested paths can improve video quality. This is so because the SDN controller has the global view and knowledge of the underlying forwarding devices (e.g., packet loss and available bandwidth). The SDN architecture accelerates network innovation by using the Southbound API called OpenFlow protocol [67] which is an interface between the control plane and the forwarding plane. The Northbound interface such as RESTful API [126] Web services connects the control plane and network applications.

The OpenFlow protocol is an open interface protocol used by the SDN controller to control the forwarding devices. The SDN controller is commonly implemented remotely on a PC securely connected to forwarding devices. The forwarding devices hold flow tables which contain, packet headers for matching incoming packets, list of actions which must be followed to handle matched packets and collection of statistics for each flow such as number of packets and bytes received and transmitted [26]. Figure 3.1 depicts the packet flow processing along with the mechanics of matching and action handling through an OpenFlow.

OpenFlow ports represent network interfaces that logically connect OpenFlow switches for processing packets. Packets that are transmitted through OpenFlow switches are received on tan ingress port and processed by the OpenFlow pipeline that makes decision to forward packets to the output port. The OpenFlow pipeline indicates the set of linked flow tables that provide matching, forwarding, and packet modification in an OpenFlow switch. When the packets enters the table, a set of actions associated with that packet can be applied and executed. The action set may instructs the packet to exit the processing pipeline. It is worth mentioning that, the OpenFlow pipeline of every switch in SDN contains multiple flow tables consisting of multiple flow entries. As shown in Figure 3.1

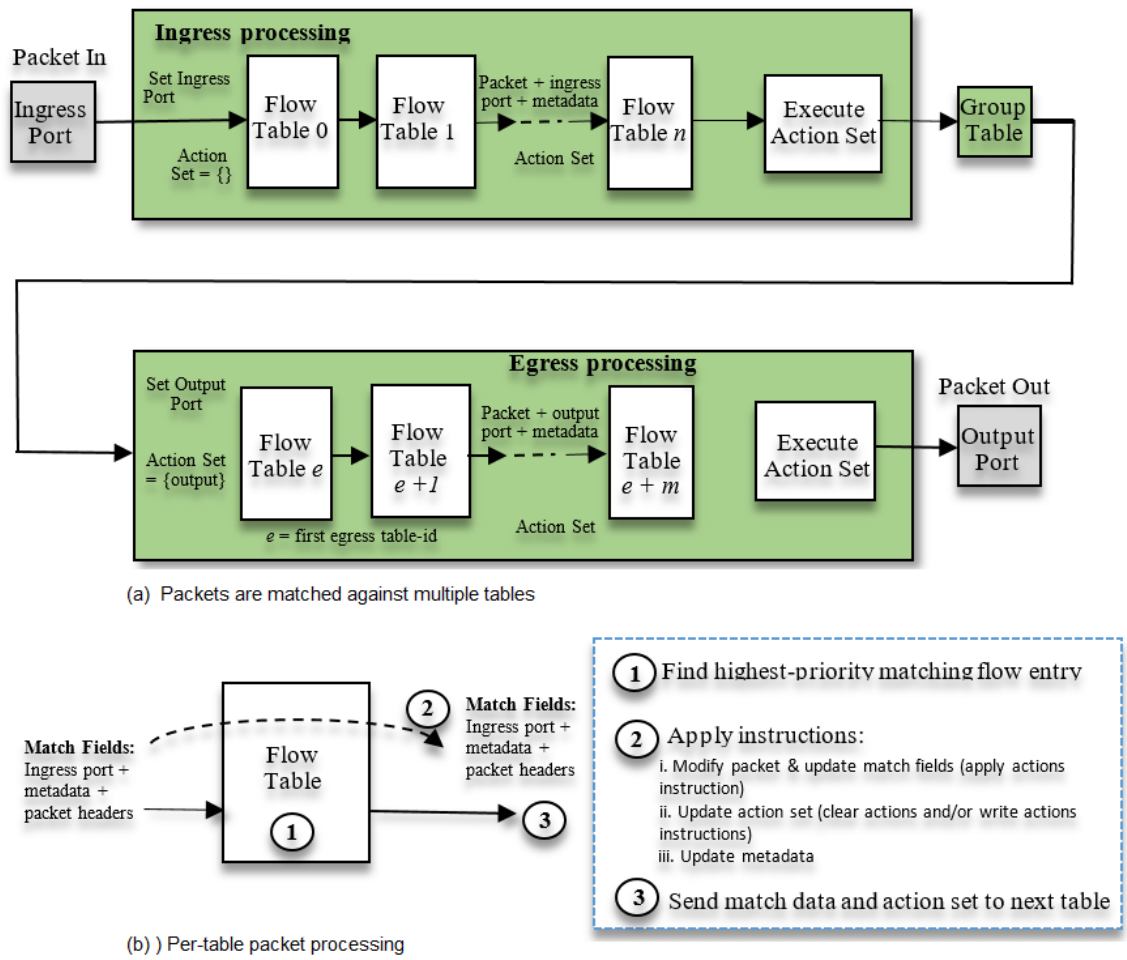

Figure 3.1: Packet flow through an Openflow switch [4]

(a), the OpenFlow pipeline processing defines how packets interact with those flow tables.

Figure 3.2 shows the functions that are performed by an OpenFlow Switch when a packet is received in a flow table. The switch uses a packet match fields (e.g., Ethernet source address or IPv4 destination address) to perform a table lookup in the first flow table. The packet is matched against the flow table based on the fields defined in the flow entry table for the lookup. When the packet match occurs, the packet with highest priority flow entry is selected and the instruction set included in the packet is applied. Every flow table in SDN OpenFlow switch supports a table-miss flow entry in order to process table misses. The table miss flow entry must specify how to handle packets that are unmatched, and the forwarding device may decide to send packets to the controller, drop them or direct them to a subsequent table [26]. As presented in section 2.2, DASH is an adaptive bitrate streaming approach that enables high quality streaming of media content over the

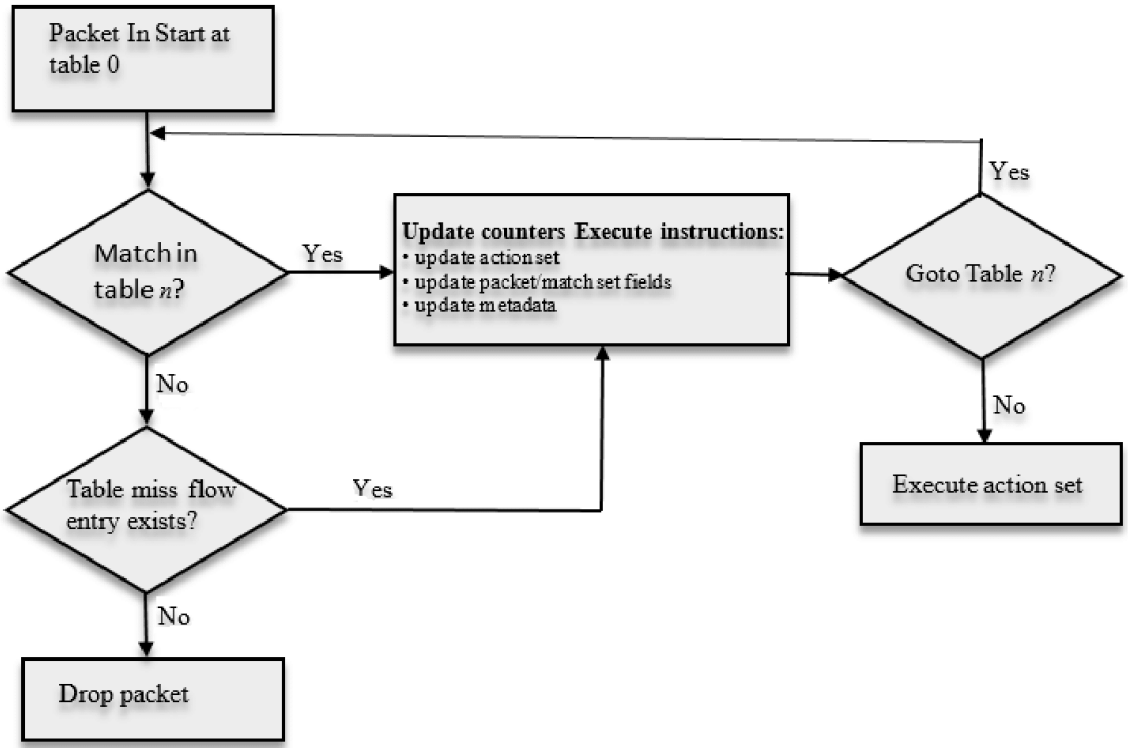

Figure 3.2: Packet flow processing and action handling in OpenFlow [4]

Internet. However, the one drawback of DASH is the frequently bitrates and resolution switching as shown in [127]. Huang *et.al* [128] also demonstrate that, DASH performs poorly under the presence of bursty traffic. To mitigate the disadvantages of DASH, video quality management scheme based on the traffic intensity could be used to complement the DASH in a scenario whereby all the SDN network flows are congested. The aim of this chapter is to propose a video quality management scheme over an SDN architecture in order to reduce the number of stalls and their duration. The structure of this chapter is organised as follows: section 3.2 presents related work followed by a description of the proposed video quality management algorithm in section 3.3. Section 3.4 provides the performance analysis and evaluation of the proposed scheme followed by a summary of this chapter in section 3.5.

3.2 Related Work

SDN performance studies have been conducted in the past using both simulation experiments and analytical modelling. Palma *et.al* [129] propose QueuePusher, an approach that efficiently handle the management of Traffic Control Queues in OpenFlow switches. Kleinrouweler *et.al* [102] uses SDN controller to provide signaling feedback on target bitrates

to DASH players in order to maintain stable video quality under the SDN architecture and reduce frequent bitrate and resolution switches. Jarschel *et.al* [130] propose a model to estimate the packet loss probability based on the queuing theory to capture the packet delay caused by the processing in the SDN controller. Although the analytical model results were successfully validated by simulation based on OMNET++, the model did not capture the delay and packet loss caused by the ability of the control plane to modify network flows within a forwarding device or among the connected forwarding devices. Liotou *et.al* [131] propose QoE SDN-based architecture to guarantee the QoE level for on demand services of OTT applications by monitoring network parameters using an SDN controller. Although authors also propose the mapping of QoS parameters to QoE but the architecture proposed was neither simulated nor implemented in SDN emulators. A network visualization and performance prediction (NVPP) tool for SDN based on queuing theory analytical models is proposed in [132]. The proposed tool can predict network performance arising from traffic variations and a real time view of a network using SDN. However, the proposed model has not been validated by either simulation or real world experiments. Hu *et.al* [133] investigate the scalability of the SDN control plane using three SDN structures of the control plane (e.g., centralized, decentralized and hierarchical structures).

3.3 The Proposed Video Quality Management Algorithm over SDN

Traffic intensity or port utilization of an Openflow switch can be implemented to avoid the queuing overflow that may cause packet loss and therefore affect the quality of transmitted videos. Based on this approach, a video quality management scheme is proposed [26] based on traffic intensity over an SDN architecture. The main idea is to investigate the performance of the SDN control plane towards contributing to the delivery of a stable and high quality stream by reducing the number of stalls and their duration. The network flow switching scheme is triggered by the virtual port queue congestion which can occur in any of the virtual ports connecting the virtual switches and hosts. The proposed video quality management scheme is able to collect statistics from virtual ports of SDN network. It uses the collected statistics to avoid congestion by modifying the traffic flows on the overloaded links due to queuing delays in the network. When congestion occurs in the

Table 3.1: Summary of Notations and Parameters used for Implementation

Symbol	Descriptions
λ_{ij}	the average arrival rate of video packets at port i of SDN switch j
μ_{ij}	the average service rate of video packets at port i of SDN switch j
ρ_{ij}	the maximum queue size of port i of switch j
p_{ij}	the video packet size that arrives at port i of switch j

virtual ports connecting the SDN switches and the hosts, the scheme is invoked to direct traffic flows to less congested forwarding paths. Table 3.1 shows the parameters used in the proposed scheme. The λ_{ij} and the μ_{ij} are calculated based on the JavaScript Object Notation (JSON) values of packets and bytes received and transmitted on each port. The average packet loss rate l_{ij} at port i of switch j is obtained through a periodic retrieval of JSON values obtained due to packet transmission errors in the network. In order to track the queue size, the traffic intensity T_{ij} that indicates a port utilization of a switch is defined using equation 3.1:

$$T_{ij} = \frac{\lambda_{ij}}{\mu_{ij}} \tag{3.1}$$

In order to avoid the queue overflow, the arrival rate should be less than the service rate such that $\lambda_{ij} < \mu_{ij}$, otherwise the queue will overflow. It is important to note that, the traffic intensity is computed at every port i ={1....n} of switch j = {1.....m}. The traffic intensity is categorized into three levels: Low, Medium and High. As such, the proposed scheme redirect the network flow which is in High level of traffic intensity to another path with Low or Medium traffic intensity if its threshold (i.e., flow_traffic_intensity $\leqslant$ T_{ij} has been reached). In order to determine the traffic intensity threshold, measurements were performed using the maximum queue size of 400 packets, the bandwidth of 100 Mbps for the selected video sequences [26]. The threshold of 0.75 was selected for High level of traffic intensity because it performed better than other thresholds under the same conditions for triggering the network flow switching. For Medium level of traffic intensity, the threshold of ($0.6 \leqslant T_{ij} < 0.75$) was selected. All other thresholds out of this range performed poor and were not selected. The video streaming is considered as the traffic to be investigated where the rest of the traffic is considered as the background traffic. As shown in Fig. 3.5, these levels are computed based on the generated TCP background traffic by hosts H2 and H3 through SDN switches S2 and S3.

Traffic flows with High level of intensity are to be directed to another forwarding path with

Low or Medium intensity. When the background traffic is high, the available bandwidth for video streaming traffic will be lower. The aim is to avoid congestion in these paths with High level of traffic intensity. The SDN controller which has a global view of the network, is used to collect statistics from the virtual switch ports. It is worth mentioning that, the SDN controller which has information of all the virtual switch port statistics use the instructions from the POST method to update the traffic flow that is congested. During video streaming, the algorithm starts with the periodic calculations of the average arrival rate and average service rate of video packets at of port i of switch j. The algorithm also calculates the maximum queue size of port i of switch j during real-time video streaming. The algorithm tracks the queue size through the traffic intensity parameter which is calculated as a ratio of arrival rate to the average service rate.When it happens that, the threshold of a traffic flow is larger than 0.75 and the flow can not be directed to less congested paths, then other adaptation techniques can be used to mitigate the network congestion. Algorithm 1 provides a summary of the video quality management mechanisms in SDN.

Algorithm 1: Video Quality Management Algorithm in SDN using Traffic Intensity

input : *Network topology information*

Start video transmission from server to client
Calculate the average arrival rate, λ_{ij}*, average service rate,* μ_{ij} *and maximum queue size,* ρ_{ij}
Compute the traffic intensity T_{ij} *based on 3.1*
if $T_{ij} \geqslant 0.75$ **then** redirect the High level of intensity to low or medium congested paths
if $(0.6 \leqslant T_{ij} < 0.75)$ **then** redirect the Medium level of intensity to low congested paths
if ($T_{ij} \geqslant 0.75$ *and the flow can not be directed to less congested paths* **then** Use other adaptation techniques to mitigate the network congestion
end if
end if
end if
Continue transmission while minimizing number of stalls, buffering events and stall duration to improve the video quality

3.3.1 Video Clips/Sequences

In order to evaluate the performance and the impact of modifying network flows on the quality of video streaming of different content types, H264 encoded video sequences shown in Figure 3.3 were used. The video sequences were categorized into slow, medium

Table 3.2: Video sequences and used parameters

Video Sequence	Average Bit rate (Kbps)	Frame rate (fps)	Resolution
Basketball	3547	30	720p
BQTerrace	3114	30	720p
Park Scene	3023	30	720p
Johnny	870	30	720p
Vidyo	890	30	720p

and fast movements. As such, the BasketballDrive, BQTerrace and ParkScene video sequences were categorized as fast movement videos because motorists, basketball players and cyclists are moving in fast motion and the camera was also in motion. The Johnny sequence was categorized as slow movement because the man is talking while seated and his upper body parts move slowly when talking. Again, the Vidyo sequence was categorized as medium movement because the men are talking while seated and their upper body parts move gently while talking.

Figure 3.3: Snap shots of the video sequences

Table 3.2, shows the video bitrates and frame rates of video sequences that were 2 minutes long each.

The DASH server for storing the video clips shown in Figure 3.3 was based on YouTube. The YouTube player API was deployed into an Apache 2 Web server in an SDN virtual host (H1). Another virtual host (H4) was used as a Web-based YouTube client. All encoded videos were uploaded into YouTube and they were re-encoded into three quality levels (720p, 480p, 360p). It is worth mentioning that, the YouTube API player was used to collect

relevant information such as the buffering duration, the number of stalls, the length and time of stalls. Figure 3.4 shows the network flow switching operation of the proposed scheme. The details of these parameters will be discussed in section 3.4 where their performance is used to evaluate the proposed video quality management scheme with the conventional DASH.

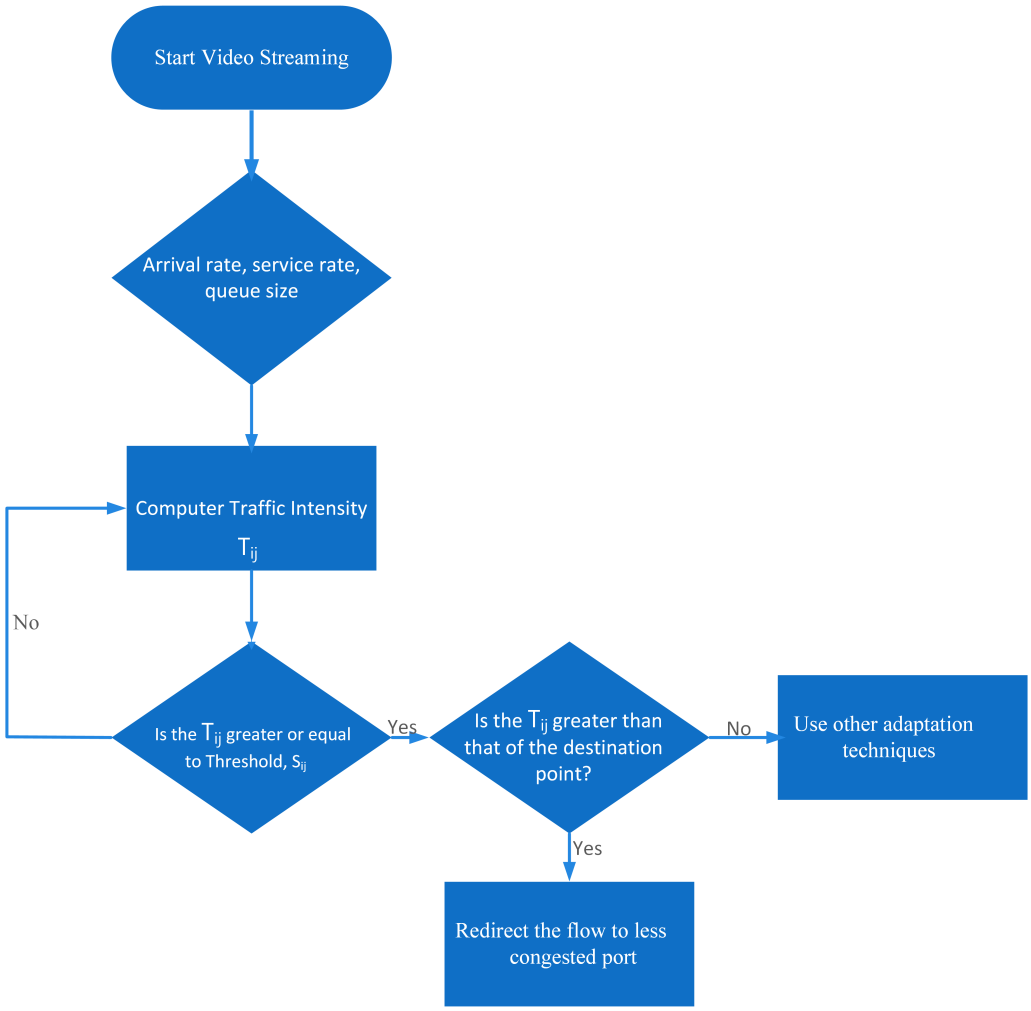

Figure 3.4: A network flow switching of the proposed video quality management scheme

3.3.2 Experimental Testbed based on SDN

To provide a feasibility study and evaluate the performance of the proposed video quality management scheme, this work employed an SDN-based platform using Mininet emulator and SDN controller. A Python based application was developed to provide an interfaces with the controller via the RESTful API Web services. The virtual switch port statistics are collected by this application. It also issues instructions on how to modify network

flows facing congestion due to queuing delay and direct them to a path with low network congestion [26] The SDN testbed is shown in Figure 3.5 where the Mininet version 2.2.0 was used to create a network of virtual hosts, switches and links. Mininet was installed in Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86 64) with 1GB of RAM and Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz. The OpenDaylight controller installed in Ubuntu 14.04.5 LTS (GNU/Linux 3.19.0-68-generic x86 64) with 2GB of RAM and Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz. was used as a SDN controller which uses RESTful API Web services to interface with network applications. The OpenFlow version 1.3 protocol was also used to provide an interface between virtual switches and the controller. In that aspect, the Mininet and OpenDaylight controller were installed in separate virtual machines by using VMware ESXi version 6 Hypervisor.

The bandwidth of network links connecting all virtual switches (S1, S2, S3 and S4) was set to 10Gbps while the bandwidth connecting virtual hosts and virtual switches was set to 1Gbps. As shown in the experimental testbed (see Figure 3.5), traffic flows can take 5 paths to arrive at H4 as follows: p_1=S1→S4 p_2 =S1→S2→S4, p_3 =S1→S3→S4, p_4 =S1→S3→S2→S4, p_5 = S1→S2→S3→S4. H4 was used as a web-based YouTube client while H1 was used as an SDN virtual host where the Apache2 Web server was installed. In addition, YouTube was used during implementation as a DASH server. The remaining hosts H2 and H3 are used to generate TCP background traffic through virtual switches S2 and S3, respectively.

3.4 Performance Analysis and Evaluation of the Video Quality Management Scheme.

Extensive experiments were performed on different video sequences for three quality levels (720p, 480p, 360p) under various maximum queue sizes. Figure 3.6 shows the average arrival rates for each quality level. The average arrival rate for fast movement video sequences (BasketballDrive, BQterrace and Parkscene) is higher for each quality level than medium (Vidyo) and slow (Johnny) movements sequences. It is imperative to note that, since the average Ethernet frame size for each sequence was almost the same at 1365 bytes, then the average service rate is approximately the same as shown in Figure 3.6. As shown in Figure 3.4, other QoE-video adaptation techniques can be used when the traffic intensity (T_{ij}) is not greater than that of the destination point. However, when the

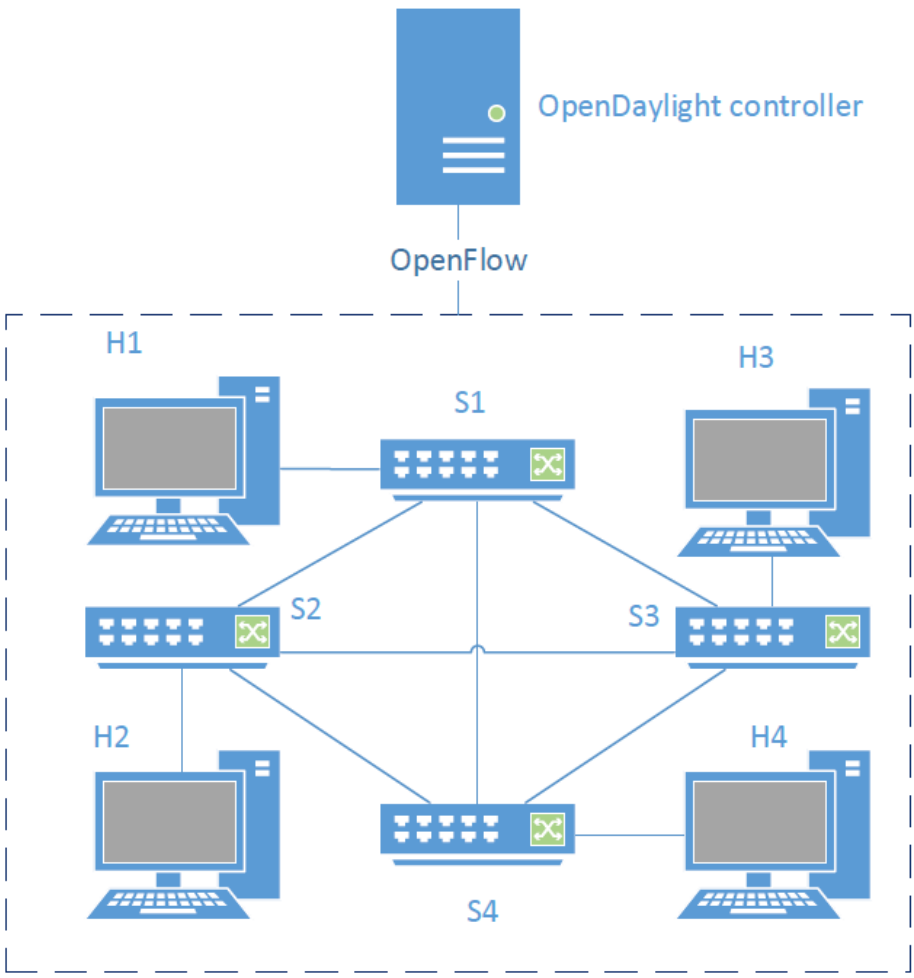

Figure 3.5: An experimental testbed used to implement the proposed video quality management scheme

traffic intensity is greater than that of the destination point then the video quality scheme is used to redirect the video flow to less congested port.

3.4.1 Implementation of DASH without the Proposed Quality Management Scheme

In order to investigate the performance of DASH without the proposed video quality management scheme, the maximum queue size was set at 400 packets whereas the bandwidth was set to 100 Mbps. Figure 3.7 depicts the number of stalls experienced by each video sequence at each quality level for high level of traffic intensity where $T_{ij} \geqslant 0.75$. As shown in the Figure 3.7, fast movement video sequences at 720p quality level experiences more number of stalls (i.e., BasketballDrive = 15, BQTerrace = 12 and Parkscene = 13) compared to medium (Vidyo = 4) and slow (Johnny = 2) video sequences. This is so because fast

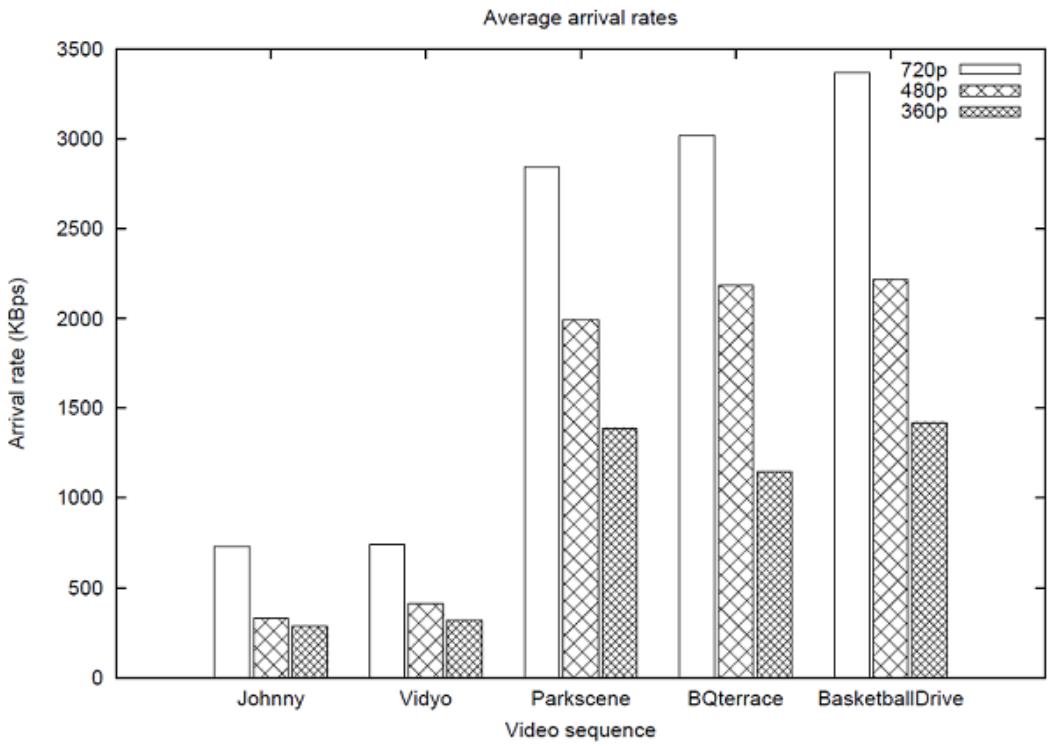

Figure 3.6: The average arrival rates of video sequences

movement video sequences occupy more bandwidth than medium and slow movements videos due to high bitrates. It is worth mentioning that similar behavior and performance can be observed at 480p and 360p video resolutions.

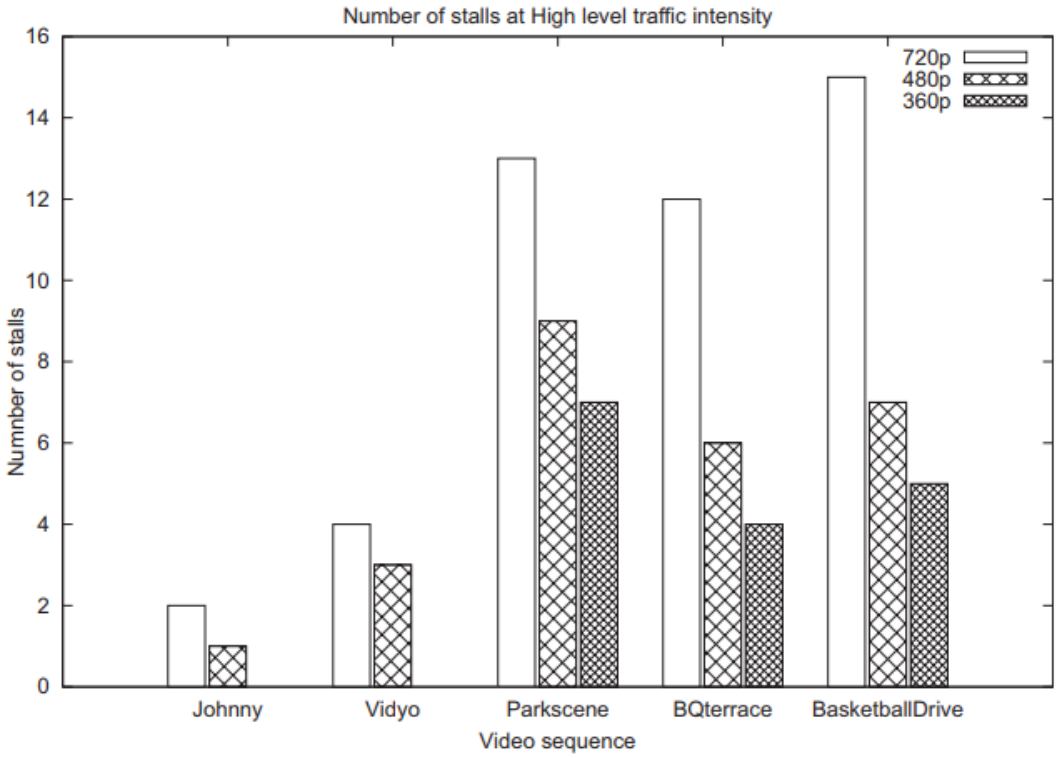

Figure 3.7: Number of stalls for high level traffic intensity

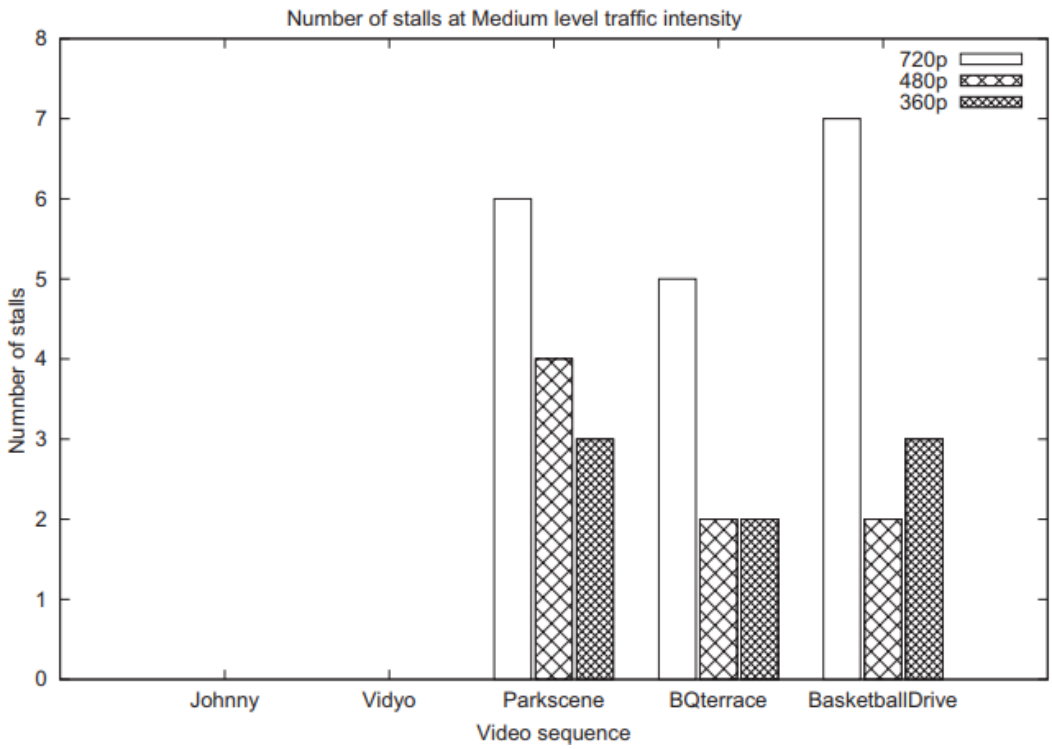

Figure 3.8: Number of stalls for Medium level traffic intensity

Figure 3.8 indicates the number of stalls experienced by each video sequence for the Medium level traffic intensity under the threshold of ($0.6 \leqslant T_{ij} < 0.75$). This is the threshold for Medium level of intensity obtained in the measurements using the maximum queue size of 400 packets and the bandwidth of 100 Mbps for the selected video sequences [26]. The additional of 0.6 threshold was selected to represent the limit of traffic intensity for Medium Level of intensity during video streaming. The number of stalls for the Medium level of traffic intensity were less than that of the High level of traffic intensity. This is so because the available bandwidth is bigger than that of the High level traffic intensity. Note from Figure 3.8 that, no number of stalls reported for medium and slow movement video sequences.

Fig.3.9 illustrates the average stalling duration in seconds for High level of traffic intensity (for all video sequences) at each quality level. Fast movement video sequences experienced longer stalling time (BasketballDrive = 23 seconds, BQTerrace = 17 seconds and Parkscene = 20 seconds) than medium (Vidyo = 9 seconds) and slow (Johnny = 5 seconds) video sequences. This is so because, fast movement video sequences consume higher bandwidth than medium and slow video sequences. Note that, similar performance behavior can be observed for 480p and 360p quality levels.

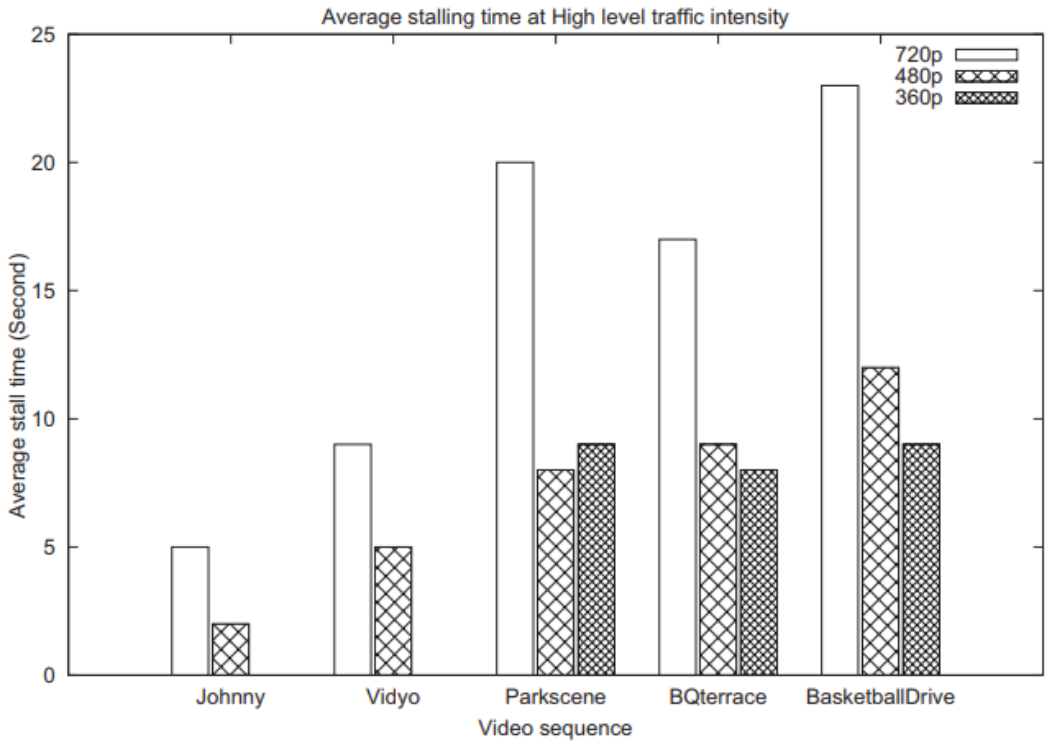

Figure 3.9: Average number of stalling duration for High level traffic intensity

3.4.2 Implementation of DASH with the Proposed Quality Management Scheme

To overcome such a high number of stalls for DASH video with high and Medium level traffic intensity shown in Fig. 3.7 and Fig. 3.8 that affect the video quality, the proposed video quality scheme [26] was deployed to redirect the network flow with High traffic intensity to an OpenFlow switch port with Low level of traffic intensity. That way, the network congestion is avoided while at the same time improving the end-user's video quality. In order to demonstrate the performance of the proposed video quality scheme over SDN, the traffic intensity threshold of 0.75 was empirically selected for High level of traffic intensity. This threshold value performed better than other thresholds under the same maximum queue size of 400 packets, the bandwidth of 100 Mbps and the selected video sequences. As shown in Figure 3.10, the number of stalls were only reported for fast movement video sequences (BasketballDrive = 2, BQTerrace = 1 and Parkscene = 1), and only at 720p quality level. The results indicate that the number of stalls in the proposed video quality management scheme is significantly smaller compared to DASH without the proposed scheme as shown in Figs. 3.7 and 3.8. While that is the case, there were no number of stalls reported once the DASH video streaming was in the new path with the Low level of traffic intensity. For the purpose of demonstrations, High level traffic intensity was only used to compare the performance of DASH with and without

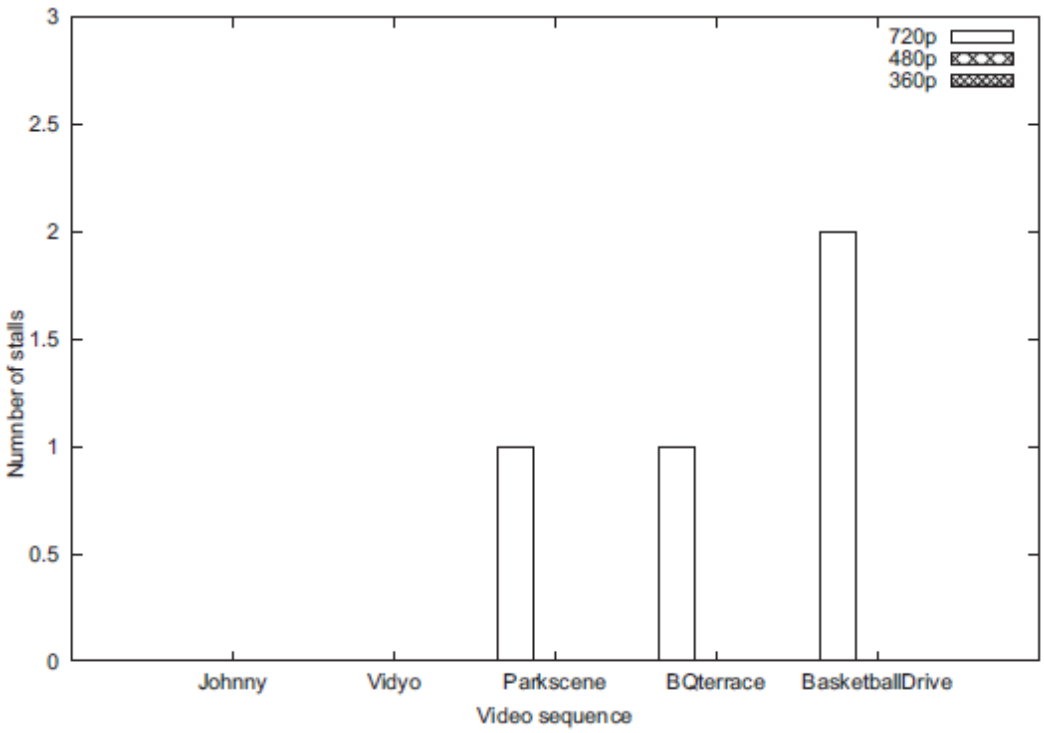

Figure 3.10: The number of stalls in the proposed video quality management scheme

the proposed video quality scheme.

To add-on, Table 3.3, shows that the number of stalls were significantly reduced by more than 84% at High level of traffic intensity for all quality levels. That way, the end-user's video watching experience is significantly improved.

Table 3.3: Comparison of number of Stalls of DASH with and without the proposed scheme

	Without the scheme			With the scheme			Reduced (%)		
	720p	480p	360p	720p	480p	360p	720p	480p	360p
Basketball	15	7	5	1	0	0	93.33	100.00	100.00
BQTerrace	12	6	4	1	0	0	91.67	100.00	100.00
Parkscene	13	9	7	2	0	0	84.62	100.00	100.00
Vidyo	4	3	0	0	0	0	100.00	100.00	NA
Johnny	2	1	0	0	0	0	100.00	100.00	NA

Extensive results for the stall duration that plays a major role in the evaluation of video quality over DASH were collected to investigate the performance of the proposed scheme. Table 3.4, shows the comparison of average stall lengths in seconds for DASH with and without the proposed scheme. It is imperative to note that, at High level of traffic intensity for all quality levels, the proposed video quality scheme reduces significantly the average stall duration by more than 94%.

Table 3.4: Comparison of DASH with and without the proposed scheme

	Without the scheme			With the scheme			Reduced (%)		
	720p	480p	360p	720p	480p	360p	720p	480p	360p
Basketball	23	12	9	1	0	0	95.65	100.00	100.00
BQTerrace	17	9	8	1	0	0	94.11	100.00	100.00
Parkscene	20	8	9	1	0	0	95.00	100.00	100.00
Vidyo	9	5	0	0	0	0	100.00	100.00	NA
Johnny	5	2	0	0	0	0	100.00	100.00	NA

3.5 Summary

To summarize, the proposed video quality scheme shows to be an effective strategy that can overcome the shortcomings of DASH approach which is attributed to frequent bitrates and resolution switches under bursty background traffic. There were no number of stalls reported for 480p and 360p quality levels using the proposed video quality scheme which indicates a significant performance compared to that of DASH technique. It is worth noting that, the number of stalls and their duration were significantly reduced by more than 84% and 94%, respectively for the High level of traffic intensity at 720p quality level. While the above strategy only improve the video quality using a single path for transmission, the next section presents a novel QoE-driven SDN based resource allocation mechanism approach over SDN/NFV networks.

Chapter 4

QoE-Driven SDN based Resource Allocation Mechanism using Network Softwarization

Network softwarization [60] and virtualization using SDN and NFV is expected to impact several aspects of network development and services such as CDN or video accelerators [61], [62]. Network softwarization using SDN and NFV enables distributed virtual platforms to execute any network functions and networked services as applications on VMs which are allocated, managed and moved dynamically on general purpose hardware. Network softwarization and virtualization as described in section 2.4 are considered as a key enabling technologies and an appealing solution in future networks such as 5G for the value added QoE-driven services provisioning such as high quality 3D/4K/8K video and high bitrate contents distribution across the network coupled with personalized services interactivity.

Chapter 3 presented the video quality management scheme based on traffic intensity over SDN to reduce the number of stalls and their duration. However, enabling the dynamic configuration and resource allocation using network softwarization can be significant and an outstanding approach on how resources are used for the purpose of improving the end-users' QoE. Dynamic resource allocation in SDN involves an allocation of resources to users' applications in the data plane by abstracting the low-level components. It also involves guaranteeing the fair allocation of such resources, transparency, and scalability to the end users. The main objectives of dynamic resource allocation in the context of virtualized based SDN networks are overall cost reduction, for instance, by improving resource usage (link and switch), and reducing the management overhead for service and content providers. Efficient techniques for resource allocation should address quality metrics such

as QoE towards resource utilization, cost, memory and power consumption reduction. Motivated by the incentives and opportunities provided by network softwarization, this chapter presents, a novel QoE-driven resource allocation strategy that dynamically assign tasks to virtual network nodes in order to achieve an optimized end-to-end quality. The main idea is to find the best combination of network node functions that can provide an optimized level of QoE to the end users through node cooperation. The task-level scheduling mechanism is designed to assign tasks related to multimedia application to different network elements. The rest of this chapter is organised as follows. Section 4.1 presents related work. Section 4.2 provides an illustration of a system overview and task assignment model followed by the description of the network model and a utility function in section 4.3. Section 4.4 presents the QoE-driven resource allocation and task assignment algorithm. Section 4.5 presents the performance evaluation and analysis of the results. Section 4.6 summarizes this chapter.

4.1 Related Work

The network resource allocation is a major problem in multimedia communications because there are dynamic changes of the service requirements (e.g. delay, packet loss, etc.) in real-time applications such as video streaming. Initially, the QoE-based optimization solutions were proposed for elastic application such as file transfers to capture the user's satisfaction as a function of data rate using a concave utility function [134]. From network operator's perspective, the aim is to keep a good level of service quality and allow the maximum number of users to join the system. With a focus on throughput maximization, authors in [135] address the optimization of network resource allocation for wireless video delivery. Abuteir *et.al* [136] propose an SDN-based framework for dynamic traffic shaping on home network gateway. Authors uses the collected network statistics and monitoring of video flow to achieve dynamic allocation of bandwidth for each video flow in real-time.

Diorio *et.al* [137] propose an SDN-based gateway that acts as a component of the OpenFlow controller to enhance the multimedia content delivery. Using the information collected in the network using an OpenFlow controller, traffic flows can be routed through different paths in the network. Jin *et.al* [138] propose a two-step iterative approach that jointly allocate resources and route video flows in order to optimize video distribution over future internet. An NFV-based routing policy is introduced first to maximize the total

cache hits by allocating storage and computing resources. In the second approach, a SDN-based routing mechanism that can configure the routing matrix for a given resource placement is proposed to minimize the networking cost. Authors formulate the joint optimization problem as a convex optimization problem using a delay utility function to represent the user's QoE. In order to maximize the revenues of ISP, Bagci *et.al* [139] propose a queue-allocation optimization solution for adaptive video streaming over SDN. These approaches indicate to be very potential in improving the end-user's QoE. However, none of these solutions consider the cooperation among network elements in improving the quality of the transmitted video flows from source to destination.

4.2 System Architecture Overview and Task Assignment Model

Media flows from the server to clients in the traditional systems follow the same network path while it is the case that, such a path might not be the optimal one for all types of media flows [112]. Figure 4.1 shows an example of an adaptive media streaming that involves different tasks. To maximize the end users' QoE, it is therefore important to develop strategies that allow delivering each user media flow using the "best available" path. Motivated by the Economic Traffic Management concepts presented in [140] this chapter provides a novel QoE-centric traffic flow control and routing approach. The chapter showS that SDN/NFV is a key factor for enhancing video quality, resource allocation and QoE management in future networks as shown in Figure 4.2. The QoE-based resource allocation and configuration is deployed in the SDN controller of the control plane. Specifically, the presented QoE-centric traffic flow control and routing mechanisms enable network elements to cooperate in measuring and collection of the QoS/QoE influencing factors in the SDN network. Figure 4.3 illustrates components of task assignment model that provides QoE-driven resource allocation in SDN/NFV. As shown in the application system overview, the workflows w_i contain video streams with different independent jobs. The video streams consist of varying video resolutions. The considered video streaming service can be decomposed into a set of tasks. As shown in Figure 4.3, a set of tasks are described as a Directed Acyclic Graph (DAG) of tasks denoted as: $G_T = (T, E_T)$. $T = \{1, ..., \lambda, ..., \Lambda\}$ represents the set of tasks and $E_T = (e_{vw})$ is the set of edges such that each edge e_{vw} represents a unidirectional data transfer from task v to task w. The task model considers a video streaming service which includes four tasks:

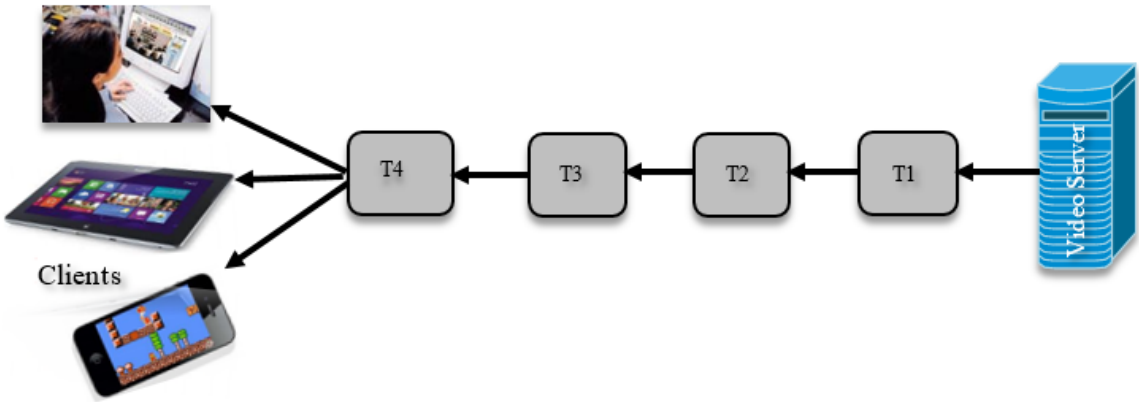

Figure 4.1: An example of a DAG for adaptive media streaming

caching (original video server), encoding, forwarding and playing back (client side). The forwarding task refers to one forwarding action running in one network node. As shown in Figure 4.3, the system consists of task dispatcher, task mapper to network elements, resource manager and the task assignment algorithm described in section 4.4. The task dispatcher is used for dispatching jobs and tasks from the media server based on the client's requests. The task mapper is responsible to map tasks to NEs while the resource manager to manage network resources in network.

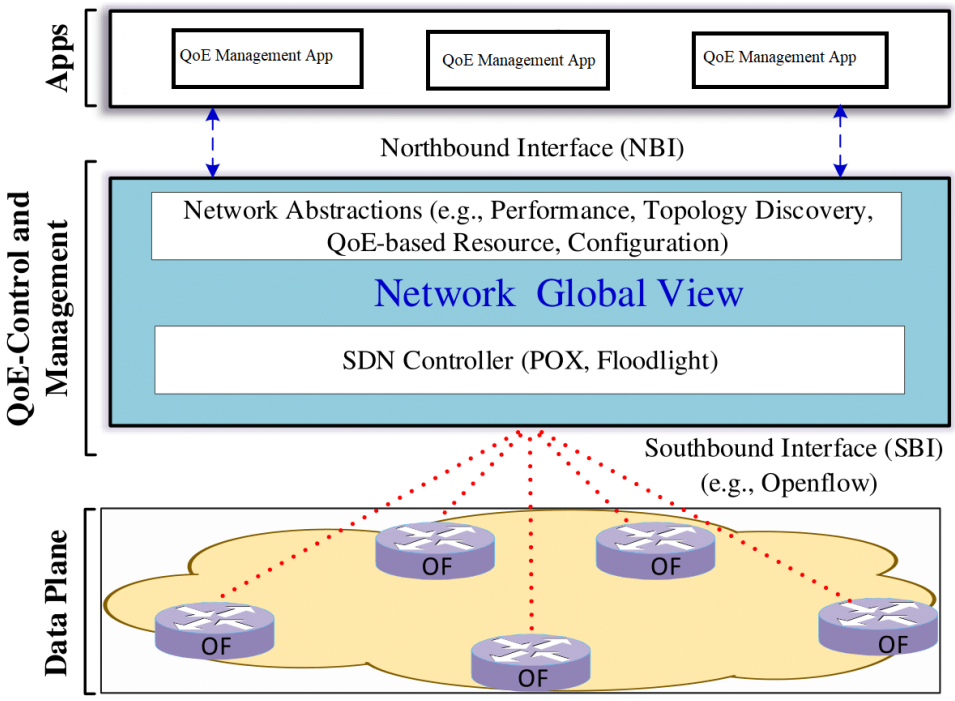

Figure 4.2: QoE-driven SDN based resource allocation system overview

A binary vector $X_i = [x_{i\lambda}]$ for $\lambda \in T$, can be assigned to each node i in the network where $x_{i\lambda}$ is a boolean value that indicates the current state of the node i corresponding to task λ. It is worth mentioning that, when node i performs a task λ then $x_{i\lambda} = 1$. Figure 4.1 illustrates

the DAG of the media streaming service task chain. Thus, each task have to be executed in order to deliver the video from the media server to the client. Depending on the network and application parameters, the aim is to optimize the overall end-user's QoE by making the best possible task assignment to network nodes. Similar to the illustration of Figure 2.5, the data forwarding plane consists of hardware and virtual resources. The hardware resources consist of compute, storage and network modules. Essentially, these are the physical resources which are related to memory, network and CPU. The virtual resources consist of the vCompute, vStorage and vNetwork modules. The VNFs are implemented in the data plane and controlled by the SDN-Controller through a Southbound API using the OpenFlow protocol. The QoE monitoring and management of media flows is done by the Management Plane via a Northbound API. The hardware resources and VNFs are abstracted using the virtualization layer (refer to Figure 2.8).

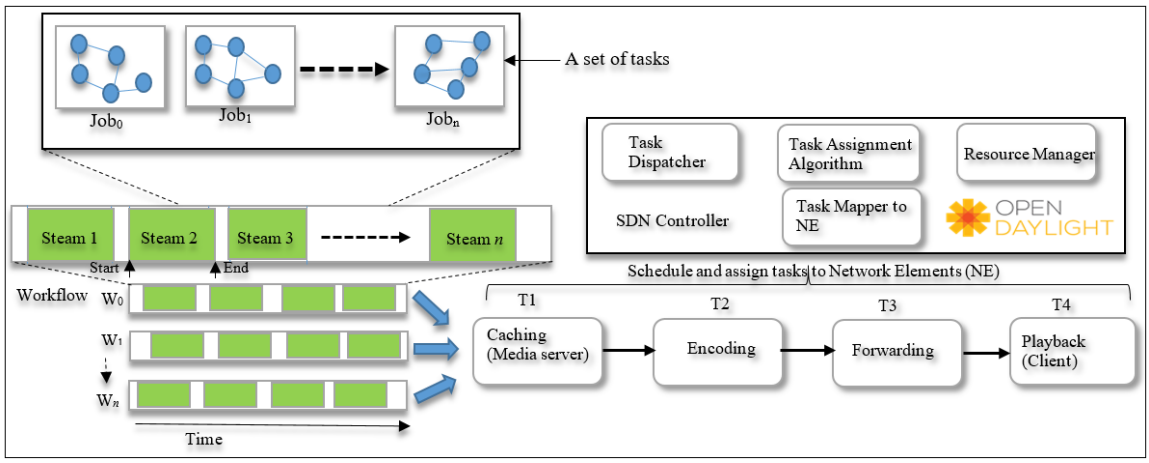

Figure 4.3: Illustration of task assignment model

4.3 Network Model and the Overall Utility Function

It is worth noting that, different services have different QoS/QoE requirements based on various parameters. Depending on user's QoE requirements, each service can be divided into smaller tasks that can be assigned to different NEs, for example different VNFs). The aim is to deliver the video to the end-user using an Openflow-based virtual switch from a media server where the original video is stored. The network is modeled as a DAG $G_Z = (Z, E_Z)$. The vertices represent the nodes $Z = \{1, ..., i, .., N\}$ and the links are described by the set of edges $E_Z = (e_{ij})$. Each node of the DAG is a Network Element (NE) which can be based on NFV where each NFV includes many VMs (e.g., for storage and network).

A utility function is defined that can formalize the correlation between network performance and user perceived quality. The concept of utility function is adopted from economics. That way, it provides a means of reflecting to normalized and transparent way of various services performance prerequisites, degree of users' satisfaction, different types of networks diverse resources. It also provides a dissimilar QoS provisioning mechanisms and capabilities under common utility-based optimization problems as demonstrated by [141]. Our algorithm decides which particular NE should execute a given task λ by maximizing network utility function.

The overall utility function consists of both the benefit and the cost for a node $i \in Z$ executing a task $\lambda \in T$. The objective function is defined as:

$$u_{net} = max \sum_{i \in Z} \sum_{\lambda \in T} (\alpha \times b_{i\lambda} - \beta \times c_{i\lambda}) \times x_{i\lambda} \tag{4.1}$$

where α, β are the weighting factors. $x_{i\lambda}$ is a boolean variable that can be 0,1 depending if a node i executes a task λ. $b_{i\lambda}$ is related to the benefit that exists, if a task λ is executed in node i. $c_{i\lambda}$ refers to the cost for node i running task λ. It is defined as the cost for resource consumption of both CPU and memory, i.e. $cost = f(CPU, memory)$ and can be calculated as follows [142]:

$$c_{i\lambda} = \gamma_i \times CPU_{i\lambda} + \delta_i \times Memory_{i\lambda}, \forall i \in Z, \forall \lambda \in T \tag{4.2}$$

where γ, δ are scale factors related to node i which allow us to weight the cost according to the required CPU and memory for a particular task in node i e.g. a task such as "encoding" needs more CPU and memory than a "forwarding" task. These weights depend on node i. Furthermore, the benefit is related to QoS level regarding to delay, jitter, bandwidth and packet loss. $b_{i\lambda}$ is defined as the execution benefit for running task λ at node i. A correlation model from [143] is used to map the QoS parameters to a QoE metric for video streaming service. The model is derived by a normalized QoS value as follows:

$$b_{i\lambda} = Q_r \times (1 - QoS(C))^{\frac{QoS(C) \times A}{R}} \tag{4.3}$$

where A is a constant relating to the video resolution class such as Standard Definition (SD) ($A = 120$) or High Definition (HD) ($A = 240$). If the subscribed service class is high, the constant A is assigned to a higher value. It means that the QoE level which the premium

service subscriber's requests is higher than normal service subscribers's in the network condition of the same QoS level. R is a constant which reflects the structure of the video frames according to the GoP (Group of Picture) length and it is defined as $R = 24$. Q_r is a constant factor that determines the overall QoE of video streaming service. Based on literature [143], the constant Q_r is set to = 0.95.

The normalized QoS value ($QoS(C)$) refers to the network performance and is calculated using Eq. 4.4. Equation 4.4 indicates the normalized $QoS(C)$ which is calculated as the sum of the QoS parameter values measured in the network layer multiplied with the allocated weights. The QoS parameters considered include: Packet Loss (PL), Packet Jitter (PJ), Packet Delay (PD) and Bandwidth (BW). Note that, the weights of QoS parameters are assigned according to the quality standard bounds and their relative importance degree are given from [143] as follows, PL 58.9%, PJ 15.1%, PD 14.9% and BW 11.1%. The normalized QoS value reflects the network condition and is calculated as follows [143]:

$$QoS(C) = PL \times W_{PL} + PJ \times W_{PJ} + PD \times W_{PD} + B \times W_{BW} \tag{4.4}$$

where $C = \{1..., i.., N\}$ is a sub-set of Z, a set of nodes involving in video delivery from the media server to the client. W_{PL},W_{PJ},W_{PD} and W_{BW} are the weights for packet loss, packet jitter, packet delay and bandwidth, respectively. Note that, the weights of QoS parameters are assigned according to the quality standard bounds and their relative importance degree are given from [143] as follows, PL 58,9%, PJ 15.1%, PD 14,9% and BW 11,1%. The weight of QoS parameters is assigned based on the quality standard bounds recommended in the standardization organizations (e.g. ITU-T) and its relative importance degree. The objective function is subjected to the following constraints.

Constraint 1: Every task λ must be executed in at least one node such that

$$\sum_{i \in Z} x_{i\lambda} \geqslant 1 \forall \lambda \in T \tag{4.5}$$

Constraint 2: Each node can execute only one task at a time

$$\sum_{\lambda \in T} x_{i\lambda} = 1 \forall i \in Z \tag{4.6}$$

Constraint 3: If node i is executing task λ then node j that is going to execute task $(\lambda + 1)$

(next task), must have a relation (link) with node i of $e_{ij} = 1$

$$e_{ij} = \begin{cases} 1 & \text{if } x_{i,\lambda} = 1 \text{ AND } x_{j,(\lambda+1)} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in Z \tag{4.7}$$

Constraint 4: Each network element has specific available resources and every task requires specific amount of resources. Thus, the available resources for each network element cannot be less than the required amount of resources.
For node $i \in Z$, the author define a set of available resources as

$$Available_i = \{CPU_i, Memory_i, \ldots\} \tag{4.8}$$

For a task $\lambda \in T$, a set of required resources is defined as

$$Required_\lambda = \{CPU_\lambda, Memory_\lambda, \ldots\} \tag{4.9}$$

$$Required_\lambda \leqslant Available_i, \forall \lambda \in T, \forall i \in Z \tag{4.10}$$

4.4 QoE-driven Resource Allocation Mechanisms in SDN

This subsection presents the QoE-driven resource allocation and task assignment algorithm that improves the delivery of video services to the end-user with high QoE. While that is the case, the optimization problem is first introduced to find the best possible path of nodes in order to improve the overall QoE. Although, this is an NP-hard problem, which could be time and energy consuming [144], a centralized optimization approach is applied with the main idea of making the best task assignment in order to improve QoE. A centralized optimization problem is defined, where a utility function, u_{net} is assigned to a network for a given strategy vector x_i. The goal of the designed system is to maximize its own utility and improve the delivery of video services with good quality to users. The solution presented in this chapter chooses a task assignment strategy x_i that maximizes utility function shown in 4.1. Therefore, a strategy x_i^* is preferred to a strategy x_i, if and

only if $u_{net}(x_i^*) > u_{net}(x_i)$.

4.4.1 The proposed QoE-driven Resource Allocation and Task Assignment Algorithm

Based on the system architecture shown in Fig. 4.3 and the network topology shown in Fig. 4.5, the algorithm finds the best path to deliver the video, while executing all the defined tasks. Every NE has specific available resources and every task requires a specific amount of resources regarding to the amount of CPU and memory. The QoE-driven resource allocation algorithm finds all the paths that can be used to deliver the video from each media server to each client and creates a list of them. Based on the proposed network model and task model the algorithm creates a new list with all possible paths considering all the constraints defined in section 4.3. Moreover, every path starts from the media server node and ends with the client node and must include nodes that execute all the tasks. The algorithm further considers different parameters such as the average delay and average bandwidth of the path along the path in the network topology. To summarize, the QoE-driven resource allocation and task assignment algorithm employs the following steps as shown in Algorithm 2: the routing metric used to determine the shortest path is the $QoS(C)$ that is calculated using equation 4.4.

Algorithm 2: QoE-driven resource allocation and task assignment scheme

input : *Network topology information*

1 *Find all shortest paths* $p \in P$ *in the network and creates a new list with all possible paths considering all the constraints*
2 *Compute link delay, bandwidth, packet loss, jitter for all paths*
3 *For every path, calculates the* $QoS(C)$ *value of the path based on Eq. 4.4.*
4 *For every path, calculates the "Benefit of the path" based on Eq. 4.3 by considering the* $QoS(C)$ *value of a path,* Q_r *(a factor which determines the overall video quality of the video streaming), the resolution class* A *and the structure of the video frames* R.
5 *For every path, calculates the "Cost of a path" based on Eq. 4.2 by considering the required amount of CPU and memory of a task* λ.
6 *calculates the* u_{net} *function, based on Eq.4.1 using the "Benefit of the path" and the "Cost of a path".*
7 *Select the path with the highest* u_{net} *value to deliver the video to the Client.*

4.4.2 Video Clips/Sequence used for Implementations

In order to evaluate our approach, the reference video file of an animated film called "Big Buck Bunny" which is widely used by researchers in the area of adaptive content

Table 4.1: Resolutions of video streams.

Video resolution	Video bitrate (kbps)
1080p	100, 200, 600, 1000, 2000, 4000, 6000, 8000
720p	100, 200, 400, 600, 800, 1000, 1500, 2000
360p	100, 200, 400, 600, 800, 1000

distribution was selected. The uncompressed YUV video files in 360p, 720p and 1080p resolution were then encoded using the H.264 codec. As shown in Table 5.1, the resolutions

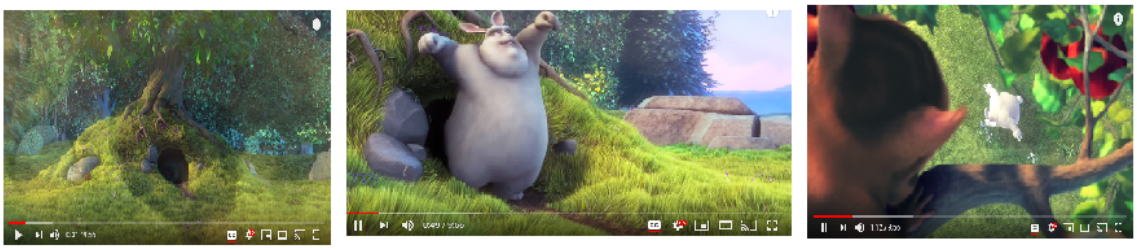

Figure 4.4: Video clips/sequence

of video streams to be delivered to clients were selected randomly between 360p, 720p and 1080p in the beginning of the experiments which was defined with the duration of 10 minutes and three clients selected randomly to receive a video from video server 1.

4.4.3 Experimental Testbed using SDN

An experimental testbed was setup based on a network emulator, Mininet [145] and an SDN-Controller implemented by OpenDaylight [146]. The testbed consists of 2 video streaming servers, 3 clients, 7 virtual switches and the SDN controller, as illustrated in Fig. 4.5. The network access was provided by using a Cisco Linksys x1000 device compatible with IEEE 802.11b/g/n operating at 2.4GHz bandwidth. Mininet was installed in a Toshiba computer with Intel @ Core^{TM}, i7-3770 CPU@ 3.40 GHz, 16 GB of RAM installed with Linux Ubuntu 14.04, 64 bit. The SDN-controller Ethernet port was fixed to a static IP address to ensure service availability throughout the experimentation period. It is worth mentioning that, during network path assignment to a particular multimedia traffic/session flow, the proposed approach first specifies a particular network path links and nodes by utilizing an automatic optimal path configuration of each video traffic/session flows to be established through the assigned path loss probability and an average delay of the link. During video service flow establishment between a client and a

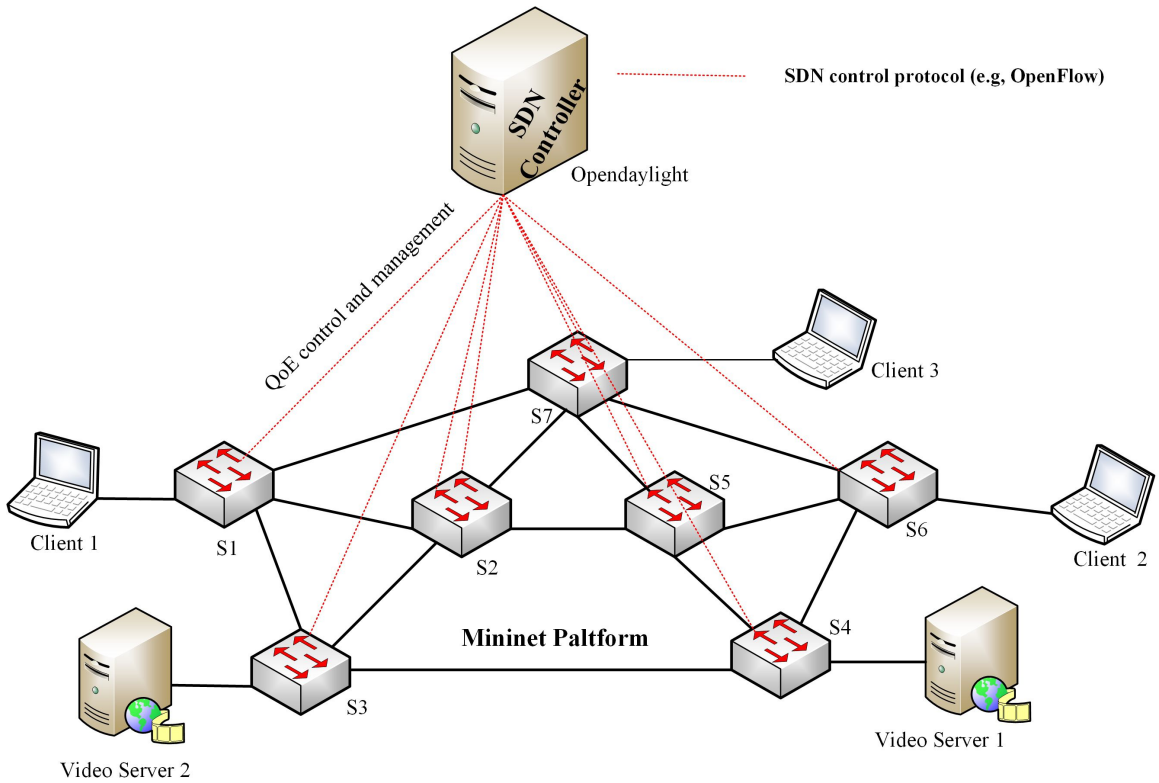

Figure 4.5: QoE-driven experimental setup based on SDN

video server, each of the following components performs the following functions:

1. End-user clients initiate a request to the video server. This request consists of client's QoE preferences and requirements based on quality parameters such as video and screen resolution as well as the supported codecs of their devices.

2. The video server(s) then communicates these clients' requests to the QoE management App with the parameter matching function. The server also conveys client's information and requested parameters such as required video resolution, video bit rates and codecs.

3. The QoE management App through the resource manager determines the required parameters and assign resources to be delivered to the client when there is a match received from the video server.

4. The associated video traffic parameters are then sent by the SDN application to the SDN-Controller. Such parameters include the video codecs and video bit rates along with the required QoE model.

5. Using the QoE-driven resource allocation and task assignment algorithms, the SDN-Controller through the OpenFlow [67] performs the QoE-centric multimedia

traffic flow control and routing mechanisms to determine the possible path that will maximize the QoE of the end-users requests.

Note that, when scalability and interoperability become the core requirements, the system is able to create a generic solution using the SDN controller and an Openflow that all together work across different service provisioning scenarios ranging from a multitude of vendors and ISPs.

4.5 Performance Evaluation and Results

The objective of this section is to present results and the evaluation based on the experiments conducted using the proposed QoE-driven resources allocation and task assignment algorithm. The results presented in this section consider the bandwidth fluctuations, end-to-end delay variations and scalability and clients differentiation. The programmability and the overall combination of features provided by SDN and NFV as well as the openness of OpenFlow [67] enabled having a fully realization of the real world networks throughout the experiments. The SDN-based experiments use the VLC media player [147] for video streaming while, for video stream delivery from any of the video server nodes to clients the UDP/RTP protocol is used. Note that, DASH was not used in this experiments. The total available bandwidth to be accessible to video streams is set on different links to fluctuate between 20kb/s to 20Mb/s. This bandwidth limit is motivated by the Wi-Fi routers using the wireless-A and wireless-G standards which can limit connection speeds with ISPs that offer 25Mb/s for fast connections.

Three different tests were carried out to evaluate the performance of our approach. The first experimental test provides the evaluation of bandwidth fluctuations and delay variations from a video server to receiving clients. In the second test, two different experiments are carried out to evaluate the effects of packet loss on video quality at different delay variations. In experiment 1, the delay was varied in the interval [20ms, 60ms] while in experiment 2, the delay was varied in the interval [10ms, 30ms]. The aim of using different delay variations in two different experiments is to investigate the effects of these delay values on the video quality. This would allow coming out with a valid conclusion on the acceptable delay limit for adaptive video streaming in SDN-based networks. The available bandwidth in these two experiments was set to 1000kbps whereas the average packet loss

probability was selected randomly in the interval [0%, 20%]. The last test evaluates the transmitted video quality as measured by the normalized QoS using Equation. 4.4. The network QoS parameters (packet loss rate, jitter, delay and bandwidth) which are related also with video quality were configured using NetEm [148], the well-known routing and traffic control feature for system monitoring, traffic classification and manipulation in computer networks.

4.5.1 Bandwidth and End-to-End Delay Variations

Figure 4.6 shows the relationship between the number of video servers that can serve a specified number of clients at different bandwidth and delay variation values. In practice, a change in the shared bandwidth will lead to a network resource reallocation process which is basically instructed by the QoE-driven resource allocation function. Every change in video quality is accounted for in the end user's QoE while the increasing number of quality fluctuations is believed to be impacted by the number of delay variations and packet loss rate on the network links.

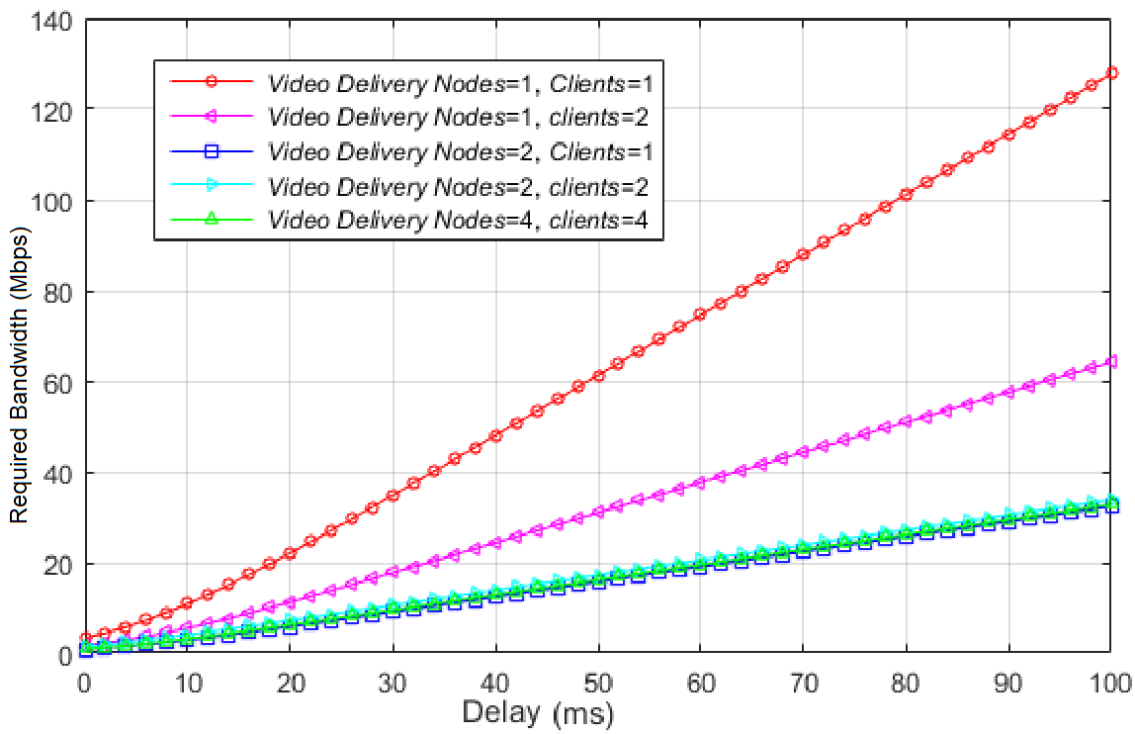

Figure 4.6: Bandwidth and delay variations with different number of clients and video servers

As shown in Fig. 4.6, as the number of video delivery nodes increases, the delay variations

on the network links have less effect on the available bandwidth required for transmitting videos to clients. For example, at 60ms delay variation, 2 or 4 video servers can provide services to 1, 2 or 4 clients at the same bandwidth of 20Mbps. Such observation has also an implication of delivering high videos quality to clients. For the video delivery node 1, the client is able to stream a video using the total available bandwidth which enable receiving high video quality compared to video delivery nodes of 2 and 4 servers.

4.5.2 Effects of Packet Loss Variations on Video Quality

As multimedia applications over IP networks continue to grow towards 5G networks, resource allocations with respect to multi-tier topology, user data sharing and cross-domain policies to be implemented using SDN/NFV strategies continue to face the challenge [149]. Considering the media streaming services in a dynamic and heterogeneous applications in future networks such as 5G, our approach differs from the conventional designs in the sense that, the utility and assignment of tasks are modelled to network nodes in order to improve the overall QoE level. Such design enables several network elements to cooperate during the process of QoE measurements and collection of the QoE influencing factors in the SDN platform. In order to do that, two different experiments were conducted to evaluate the effects of packet loss on video quality at different delay variations. In experiment 1, the delay was varied in the interval [20ms, 60ms], while in experiment 2, the delay was varied in the interval [10ms, 30ms]. The average packet loss probability is selected randomly in the interval [0%, 20%]. The available bandwidth of 1000kbps was selected based on the fact that, 1Mbps was fairly enough for our experimentation taking into account that the aim was to investigate how delay and packet loss affect the transmitted video quality using our approach. Fig. 4.7 shows the video quality of transmitted videos as the function of packet loss rate.

As expected, it can be observed from Fig. 4.7 that, as packet loss decreases, The video quality is calculated based on the mapping of QoS to QoE values using the execution benefit, $b_{i\lambda}$ for running task λ at node $i$benefit. It is worth noting that, the video quality increases as indicated by the QoE values from the correlation model described in [143].

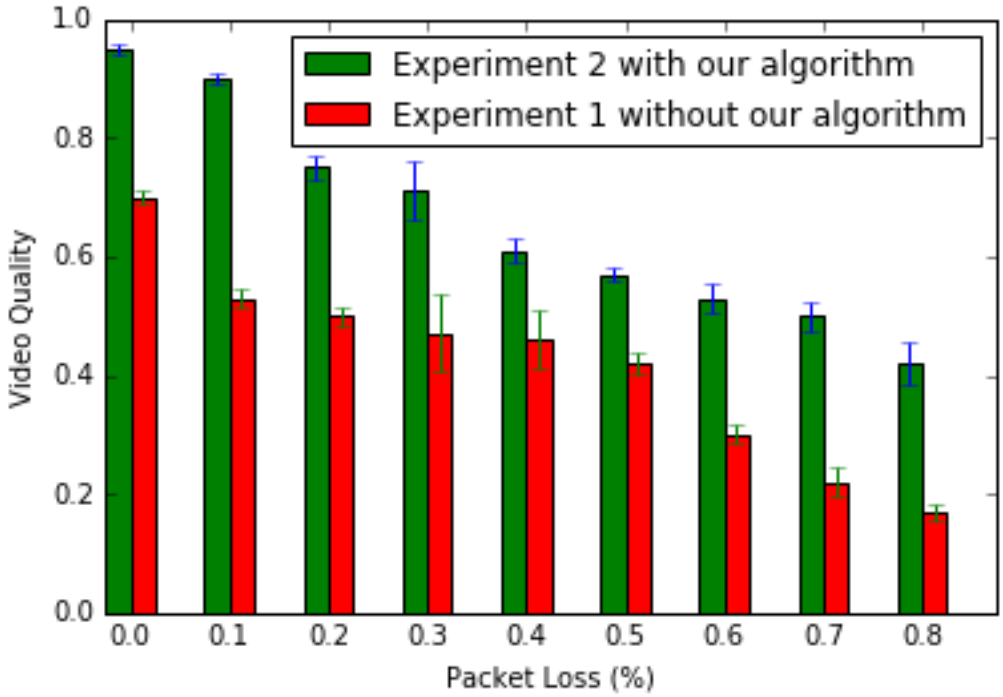

Figure 4.7: QoE values at different variations of packet loss and delay

4.5.3 Transmitted Video QoE with the Normalized QoS Values

Figure 5.6 shows the results of the transmitted video QoE without and with our proposed QoE-driven resource allocation algorithm. In this experiment, the normalized $QoS(C)$ was calculated based on Equation 4.4 where the constants A and R in Equation 4.3 were assigned to 240 and 24 respectively by considering the used codec, a set of network parameters and video resolution. The blue color bar demonstrates the test experiment without using the proposed QoE-driven resource allocation and task assignment algorithm.

It is evident that, the video quality increases as the normalized QoS value decreases. The video quality and the $QoS(C)$ reflects the QoE metric for video streaming services and the network parameters/conditions respectively which are set as described in the second test done in the previous subsection B. Using the QoS-to-QoE correlation model in [143], our proposed approach can significantly improve the video quality at the normalized QoS values.

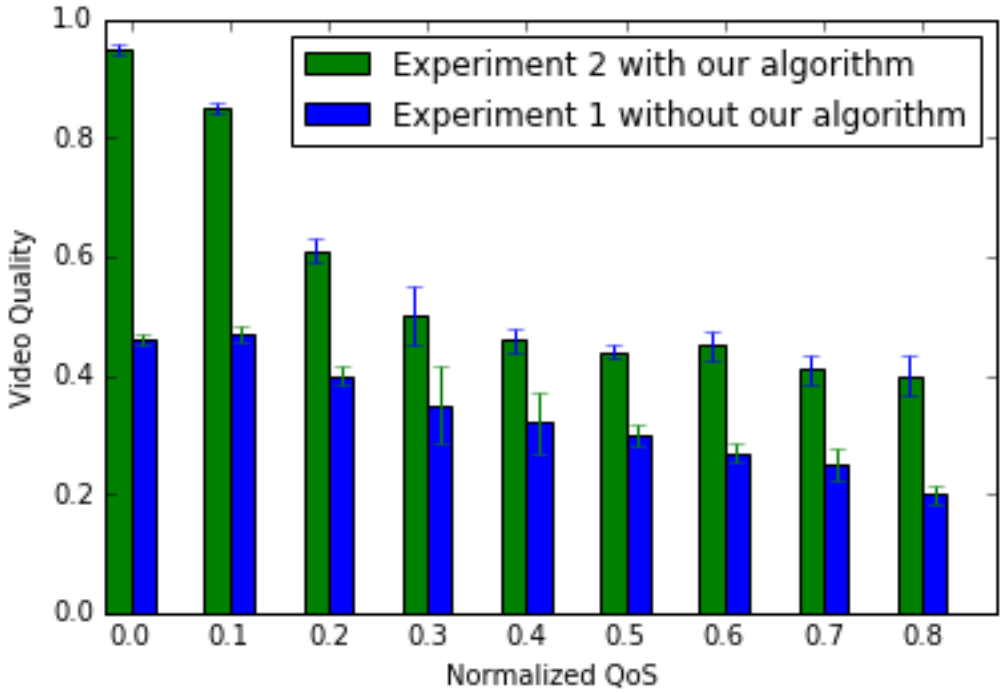

Figure 4.8: Transmitted video QoE with the normalized QoS values.

4.6 Summary

SDN and NFV promise new opportunities for offering a unified QoE control and management of multimedia services by leveraging these network softwarization technologies in future 5G wireless/mobile networks. This section presents an SDN-based strategy for QoE management by allowing cooperation and information exchange among network elements which are involved in the service delivery chain (e.g., from the video delivery nodes to clients). A novel QoE-driven dynamic resource allocation and task assignment scheme for adaptive video streaming over SDN/NFV enabled networks is comprehensively. The presented strategy enables more efficient resource utilization and simplifies network management using the elasticity of SDN/NFV technologies. This is achieved by finding and providing the best combination of network nodes that can cooperate during the execution of the defined task in order to improve the overall QoE level of the end users. It is worth mentioning that, network softwarization infrastructures using SDN/NFV is a key factor for enhancing video quality, resource allocation and QoE management especially in future multimedia networks.

Chapter 5

QoE-Centric Multipath Routing for Multimedia Services using SDN/NFV

Future networks (e.g., 5G) are expected to support and provide an end-to-end over-the-air latency of less than $1ms$, transmission reliability of 99.999% and approximately 100% services availability [150]. With the increasing number of new applications beyond personal communications, mobile devices will probably reach hundreds of billions when 5G is deployed into the market. The 5G network systems around 2020 and beyond will need to deliver as much as 1000 times capacity compared to the current commercial 4G cellular systems [151], [152]. 5G is set to provide access to any service with better quality to end-users at anytime, anywhere through reliable and cost effective communications, over any medium and across multi-operator domains using different technologies SDN and NFV. Reliable transmission protocols with high transmission efficiency in wireless environments are required in order to support a multi-variety of services (e.g, live video streaming and video gaming) in 5G networks. Requirements such as high throughput, resilience and reliability, consistency and service availability, ultra low-latency have to be archived, for 5G to support these applications [150]. Efficient transfer of large data, especially multimedia traffic flows is crucial to the performance of 5G multi-rooted topology networks where multiple paths exist between pair of hosts. In order to provide an optimal throughput for large flows and cope with the explosive growth of multimedia traffic, new transfer protocols such as MPTCP and advanced forwarding technologies such as SR [153] have been designed. The aim has been to increase network throughput, reliability, availability and make a timely response to end-users' requests as well as improving end-user's QoE.

This chapter explore the utilization of MPTCP and SR in SDN-based networks to improve

network resources utilization and end-users' QoE for delivering multimedia services over 5G networks. The rest of the chapter is organised as follows. Section 5.1 presents related work regarding multimedia transmission using MPTCP over SDN followed by a description of basic operation and principles of MPTCP in section 5.2. Section 5.3 provides the video adaptive streaming concept of MPTCP over SDN while section 5.4 presents the traffic engineering with segment routing in SDN. Section 5.5 presents a QoE-aware MPTCP SDN-based SR adaptation framework. Section 5.6 presents the proposed QoE-Centric Multipath Routing Algorithm (QoMRA) over SDN. Section 5.7 provides an experimental results and evaluation of the proposed QoE-aware MPTCP/SR-based algorithm. Section 5.8 summarizes the contributions of this chapter along with the lessons learned. '

5.1 Related Work

To pinpoint out our motivation, related work that implement MPTCP using SDN is discussed and then explore the applicability of SR for traffic engineering solutions in future softwarized networks. In [154] a proposal that uses MPTCP in OpenFlow-enabled networks to guarantee subflows of the same MPTCP connection to be sent through disjointed paths is given. A QAMO-SDN, QoS-aware MPTCP-based solution that can provide and support service differentiation in SDN optical networks is presented in [155]. The concept of SDN is invoked in [156] to establish a large-scale measurement framework on top of GEANT and PlanetLab Europe where multiple paths can be configured dynamically, and MPTCP experiments can be orchestrated in a flexible manner. A recent MPTCP-aware SDN controller that uses packet inspection to provide deterministic subflow assignment to paths and facilitates an alternative routing mechanism for the MPTCP subflows is described in [157]. Faster download speed and improved QoE is reported in [158], where SDN is used to add or remove MPTCP paths in order to reduce large number of out-of-order packets which may cause poor performance and degrade the end-users' QoE. A responsive MPTCP approach that employs a centralized SDN controller for calculating the forwarding paths of subflows is presented in [159].

An SDN MPTCP-based architecture that achieves higher data rates when transferring a huge amount of data in L2 networks is proposed in [160]. Although these SDN MPTCP-based approaches are intended to improve throughput, but they rely on installing a larger number of flow rules at the SDN switches and thereby increasing the load at the SDN

switches. Such limitation can be easily eliminated using the SR technology as shown later in this chapter. Another drawback of the above approaches, is the lack of an efficient routing and an adaption mechanism to control the number of subflows based on the network changing conditions. In addition, some of proposals do not consider on how MPTCP can improve QoE in the aspects of future IP networks for media-rich services delivery.

While recent proposals, [161], [162] attempt to achieve the optimal quality of real-time video streaming using MPTCP, authors do not consider the global control features that SDN can offer in their implementations. Such solutions are likely to suffer from congestions when large flows are transfered from source to destinations and therefore degrading the end-user's QoE. Indeed, SDN-based SR implementations can efficiently manage resources and provide better TE solutions in multi-domain 5G networks [163]. For example, an SDN-based solution for assigning TE paths with SR, based on Multi-Protocol Label Switching (MPLS) forwarding is proposed in [164] while efficient routing mechanisms for SDN with SR that meet bandwidth requirements of routing requests are described in [165]. Motivated by previous observations of SR to reduce the amount of forwarding rules in OpenFlow switches, this chapter, takes advantages of MPTCP and SR to speed up network transmission, increase network throughput, reduce overhead and further improve user's QoE for multimedia services in SDN-based networks.

5.2 Principles and Basic Operation of MPTCP

Multipath TCP (MPTCP) [161], [166] has emerged as the transport protocol capable of forwarding data traffic using multiple paths. Multi-Path TCP (MPTCP) is a recent effort of the Internet Engineering Task Force (IETF) towards standardizing a transport layer protocol that improves network connectivity and provides end-users with higher data rates. MPTCP was proposed as an extension of TCP with the main idea of splitting a large flow into multiple subflows. The subflows originating from the same MPTCP connection are then sent to their destinations through different disjointed paths in the network [154]. The receiving side that supports MPTCP aggregates the transmitted subflows and reassembles the packets that originate from different paths. As shown by Barakabitze *et.al* [166], MPTCP implementations can be integrated by new cutting-edge technologies such as SDN and NFV. MPTCP avoids network hot-spots by distributing

traffic flows over multiple concurrent paths such that all the network links are optimally loaded. MPTCP provides load balancing, security, reliable communication and better network resources utilization that leads to higher network throughput and the end-user's QoE [161], [167]. MPTCP is a promising approach to improve performance and reliability of video streaming services in future networks (e.g., 5G). The main design goals and benefits of MPTCP over TCP protocol include the following:

- ***Reliability***: MPTCP provides reliable communication because multiple paths exists that can keep the connection alive between endpoints when one path fails during data transmission. For multi-homing devices, MPTCP can also enhance reliability through the use of multiple wireless links, for example, Wi-Fi and LTE [168], [169] for data transmissions.

- ***Bandwidth aggregation***: MPTCP can potentially improve the network throughput by transmitting video flows using multiple number of available paths. That way, MPTCP efiňĄciently achieve bandwidth aggregation such that a multi-homed device can obtain a much better performance during video streaming of contents from a service provider.

- ***Fairness and Resource Pooling***: MPTCP can ensure that competing TCP flows share the available bandwidth of the shared bottleneck link. MPTCP advocates the resource polling (RP) principle by making an improved use of multiple path resources. MPTCP achieve a practical multipath-aware end-system making different paths in the network to act as if they were a single large resource.

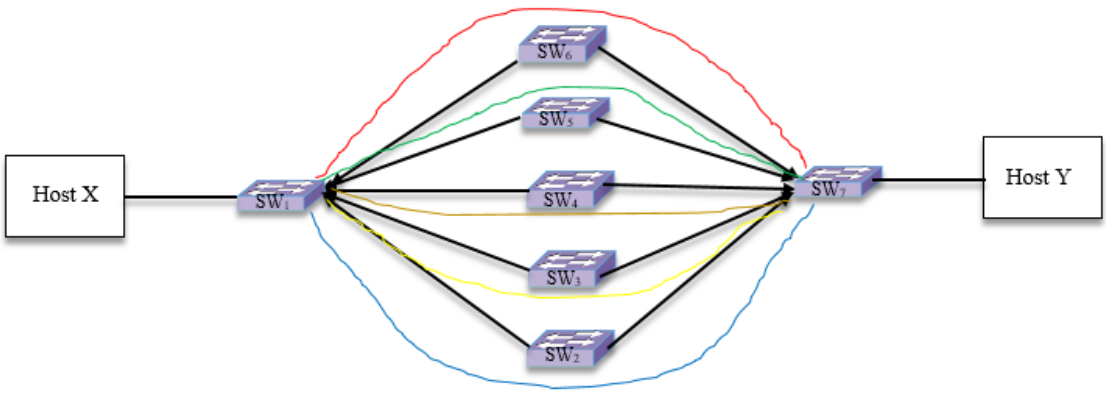

Figure 5.1: MPTCP link aggregation between Ethernet switches

Data centers and work stations can implement the concept of MPTCP in Local Area Networks (LANs) using a technique of link aggregation or Link bonding which allows

combining multiple Ethernet links into a single logical link between two networked devices as shown in Fig. 5.1, (Note that, SW = Switch). It is worth mentioning that, the aim of multipath transmission using link bonding is to make sure that multiple independent links coordinate and provide a larger bandwidth compared to the one can be offered using a single link.

5.2.1 MPTCP Connection establishment between End-points

MPTCP is implemented in the transport layer and can support both IPV4 and IPv6. The design of MPTCP follows two outstanding requirements, namely the application compatibility and network compatibility. On the application compatibility means that any application that runs over TCP should also run over MPTCP without any modifications. The network compatibility ensures that MPTCP should operate over the Internet without any need for change in networks, devices and apps. As shown in Fig. 5.2, a TCP flow connection is defined by 4-tuple (source and destination IP addresses and TCP port pairs) using a three-way handshake. A MPTCP connection is established using 7 steps as shown in Fig 5.2. Step 1-3, is a three-way handshake where a DASH client sends the SYN (for "synchronize") segment that contains the MP_CAPABLE option. This packet contains the source port and initial sequence number and is sent to the port where the MPTCP server is listening. The server replies to the SYN segment with an ACK message which includes the MP_CAPABLE option. The server also provides its initial sequence number and other supported TCP options.

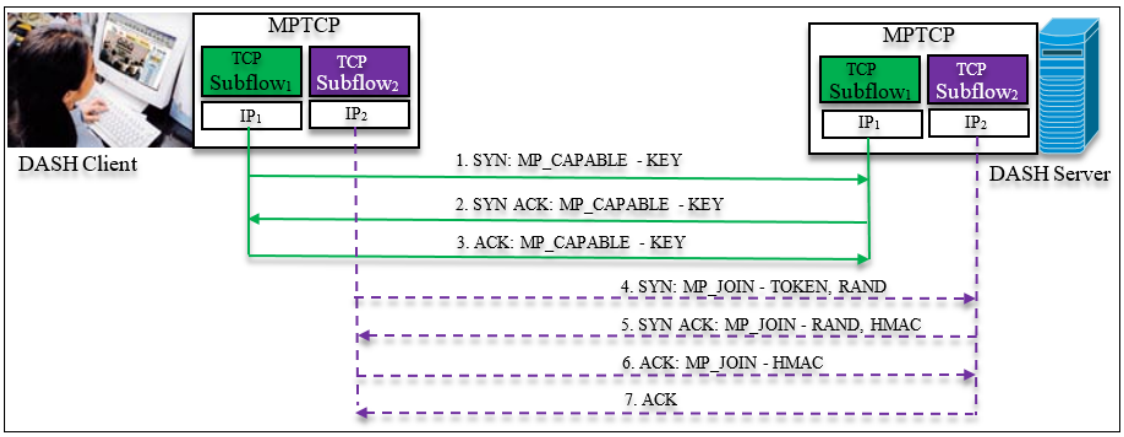

Figure 5.2: MPTCP subflow connection establishment

The DASH client then sends an ACK: MP_CAPABLE -KEY packet to acknowledge the connection establishment. For attaching the second subflow, the MPTCP client and MPTCP server then use the MP_JOIN option during the handshake (Step 4-6). Step 7 is the confirmation message containing the Hash-based message authentication codes (HMAC) ACK messages received by the DASH MPTCP client from the MPTCP server. The third subflow is attached using the MP_JOIN option exchanged during the attachment of the second subflow. After the MPTCP connection between a client and server has been established, the MPTCP default scheduler can then push data using the lowest Round Trip Time (RTT) of the subflow. Note that, subflows use their TCP sequence numbers which are in turn also used by MPTCP to make sure that, different subflows are delivered in their respective order on the receiving side. The packets are retransmitted using the next subflow when loss happens during data transmission. When data transfer is over, then TCP connection can be closed.

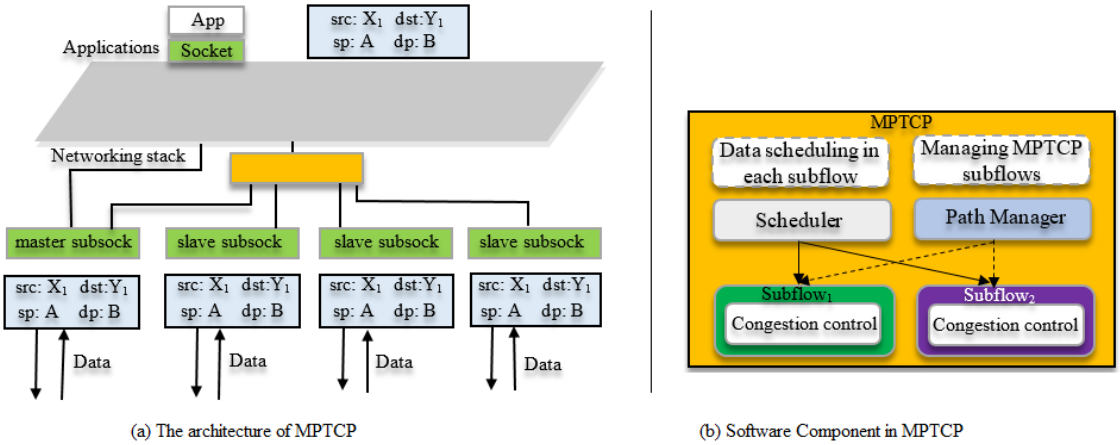

Figure 5.3: The architecture and software components of MPTCP. Note that, for compatibility, the MPTCP architecture keeps the standard socket API to legacy applications .src, dst: Source and Destination IP address, sp, dp: Source and Destination Port [5]

5.2.2 Path Management and Scheduling

Fig. 5.3 (a) shows the architecture of MPTCP where a standard TCP socket API is kept for compatibility purposes so that legacy applications can run transparently over MPTCP. Fig. 5.3 (b) indicates the three software components in MPTCP namely, *Scheduler*, *Path Manager*, and *Congestion Controls*. The MPTCP scheduler runs a scheduling algorithm that allocates traffic by selecting a subflow with the smallest RTT from different MPTCP subflows based on the available space for congestion window size. The pluggable scheduler model

presented in [170] is used in the current MPTCP Linux Kernel implementation. Three schedulers are defined, namely the (a) *around robin*, (b) *default*, and (c) *redundant scheduler*. The default scheduler basically considers the RTT and the congestion window for each transmitted MPTCP subflow. The scheduler selects subflow with the lowest RTT and which have a space for the congestion window is first assigned to packets [171]. The redundant scheduler is responsible to reduce latency for those applications consisting of moderate bandwidth requirements [172].

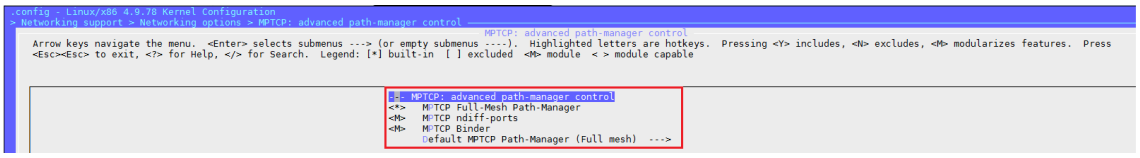

Figure 5.4: Path Manager configurations in MPTCP Linux's Kernel

A path manager entity in MPTCP is used to manage efficiently the transmission of data packets through multiple paths. Four path managers as shown in Fig. 5.4 are currently implemented by MPTCP in the Linux Kernel, namely, *default*, *fullmesh*, *ndiffports* and *binder*. The *default* path manager accept the creation of new subflows. However, it cannot initiate the creation of new subflows or even announce different IP-addresses to other subflows. The *full-mesh* creates a full-mesh of subflows among the available subflows. The fullmesh path manager is implemented in the Linux Kernel as the default MPTCP path manager such that it is able to advertise the available IP-addresses on the client side to the server and listen to those IP-addresses that are announced from the receiving side. When the interface go up or down, the fullmesh path manager can add/remove an IP-address because it is able to listen to events from other interfaces in the network. The ndiffports path manager creates multiple subflows across the same pair of IP-addresses (e.g., same source and destination IP addresses, the same destination port) but with different source addresses. The benefits of this path manager is that, it can exploit the equal costs multiple paths available in the datacentre. However, it is unreactive to network interface changes and cannot learn the available IP addresses available on the client and server side automatically. Moreover, the *binder* path manager utilizes a Loose Source Routing (LSR) mechanism [173] for distributing subflow packets.

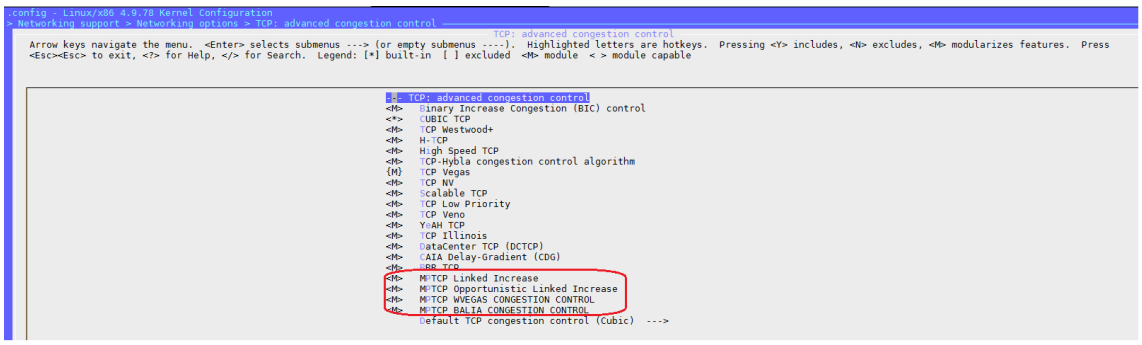

Figure 5.5: Coupled congestion control algorithms in MPTCP Linux Kernel

5.2.3 Congestion Control and Avoidance in MPTCP

MPTCP Linux Kernel implements various congestion control algorithms to improve system throughput, provide fair resource sharing and balance congestion in the network. Currently, the MPTCP Kernel implements four coupled congestion controls and avoidance algorithms, namely, Linked Increases Algorithm (LIA) [174], Opportunistic Linked Increases Algorithm (OLIA) [175], Balanced Linked Adaptation (BALIA) [176], and Weighted Vegas (wVegas) [177] on each subflow as shown in Fig. 5.5. When no packet loss is detected during transmission, the congestion avoidance algorithm of TCP consists of additive increase behavior while when packet loss is detected the multiplicative decrease behaviour is encountered. Using these congestion control algorithms, MPTCP must utilize the least congested mostly during the congestion avoidance phase. The main goal of LIA is to improve throughput and balance network congestion by simply switching traffic from congested network paths to least congested paths. OLIA's implementation departs from the responsiveness of LIA to provide an optimal congestion balancing and fairness between subflows in the network. OLIA is based on the New Reno Congestion control algorithm. It is the main coupled algorithm implemented in the MPTCP Linux Kernel to mitigate unfairness problems of uncoupled congestion control mechanisms for clients sharing the same bottleneck link.

As shown in Fig. 5.6, each path starts transmitting when the connection has been established after the three-way handshake procedure. Before entering in congestion avoidance stage (e.g., no errors occur) during transmission, it double the Congestion Window (CWND) and measures its state variables such as RTT while making sure that all ACK packets are received. The CWND of each subflow path is maintained until the end of the transferring data to the receiving side. A subflow starts and terminates as a regular

single-path TCP (SPTCP) connection. A MPTCP connection $MPTCP_R$ consists of a set of one or more subflows sf. Each subflow provides an alternative path to reacha remote end-system. The subflow's congestion window w_r increases by two terms when an ACK is received on the subflow r. The first term indicates an optimal congestion balancing defined in [174].

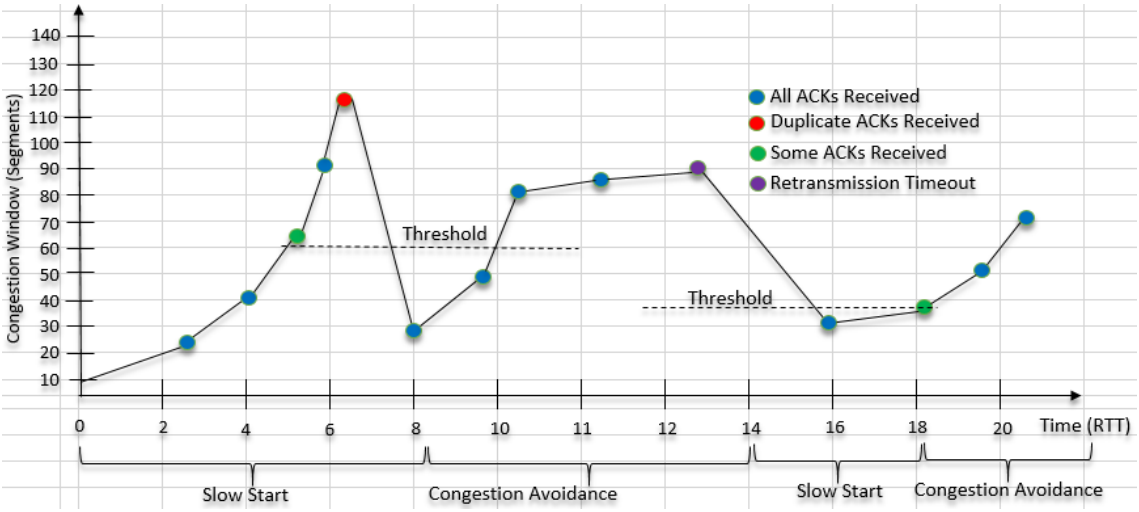

Figure 5.6: MPTCP slow-start and congestion avoidance stages [6]

5.3 A MPTCP over SDN-based Networks: Concepts and Descriptions

Despite the above mentioned benefits of MPTCP, the key technical aspects that affect the performance of this transport layer protocol is the lack of control and routing mechanisms of the splitted subflows. This is because MPTCP has no intelligent component control over the network or transmission routes used by each subflow from source to destination. In that aspect, a centralized SDN controller looks to be a fundamental tenet for routing the MPTCP subflows. Intuitively, the SDN controller that maintains the global view can provide programmable environment to implement intelligent QoE-based routing mechanisms for the MPTCP subflows [178]. However, similar to how Border Gateway Protocol (BGP) tables have grown with the spread of the Internet, Traffic Engineering (TE) implementations in the SDN data plane could results into the number of rules on every switch to grow tremendously. It is worth mentioning that, SDN switches are incapable to handle large volume of flow rules because the complex rule matching in SDN

(e.g., wildcards) requires switches to store rules in Ternary Content-Addressable Memory (TCAM), which is expensive, limited in size and needs high power consumption [179]. The limitation of SDN switches to store large number of rules can be greatly solved by SR [164] where a logical path of MPTCP subflows can be expressed as a sequence of segments between the ingress and egress network nodes (e.g., a switch/router/link). SR has been recently proposed by the IETF to provide TE by simplifying control plane where SDN switches no longer need to maintain per-demand routing information. Segment routing [163] is a new approach that greatly reduces the number of forwarding rules by encoding routing information into packet header as an ordered list of labels. SR improves the scalability of SDN since there is no path state maintenance required in each switch or router along the service delivery path.

MPTCP implementations can be integrated by new technologies such as SDN/NFV in order to achieve better network throughput, QoE-fairness in shared systems and provide flexible path allocation strategy and efficient orchestration of network resources. Consider the video streaming scenario shown in Fig. 5.7, suppose that, an MPTCP connection which consists of 3 subflows: *sf1:1*, *sf1:2*, *sf1:3* where *sf* is the subflow and the indexes indicate the number of a subflow of a particular MPTCP connection. In order to guarantee the end-user's QoE-fairness level and performance requirements of 5G networks, the SDN controller checks the available capacity of all connected paths and selects the shortest paths to transmit the subflows of the same MPTCP connection. As shown in Fig 5.7, *sf1* will be transmitted via path, S1→S2→S3→S4, *sf2* via path S1→S7→S8→S4 and *sf3* will take path S1→S5→S6→S4. The black boxes indicate video frames generated as data units by the application while grey boxes are the TCP packets to be received at the client side. One packet is lost and retransmitted.

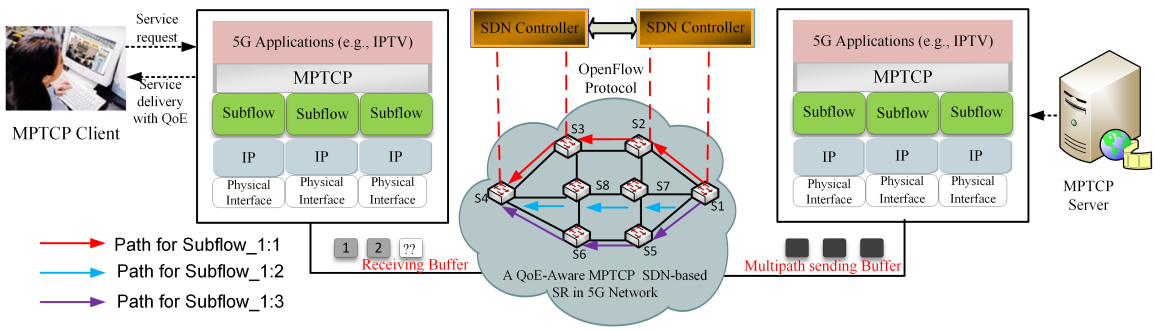

Figure 5.7: A video streaming scenario using MPTCP in SDN-based 5G networks

These MPTCP subflow paths can be expressed by the segment routing strategy into SR paths, using the segment label list. The next section 5.4 provides details of the SR

operation and the mapping procedures of MPTCP subflow paths into SR paths in the next subsections.

5.4 Traffic Engineering with Segment Routing in SDN

The primary solution used by service providers for steering the network traffic towards their destination is based on an IP lookup at each network device (switch or router) along the delivery path. While 5G networks are evolving towards QoE-based application platforms, service providers and operators are now surging to a flexible and scalable networks management through efficient routing mechanisms. Efficient transfer of large data, especially multimedia traffic flows using MPTCP and SR is crucial to the performance of 5G multi-rooted topology networks where multiple paths exist between pair of hosts. SR [163] is a source routing approach that aims to provide an advanced packet forwarding mechanism where intermediate nodes (routers/switches) are not required to install flow rules or maintain all steered paths within the network. With SR, a host or an edge router is able to steer a packet through the network using an ordered list of processing/forwarding functions called segments. A segment can be a logical or a physical element such as a network node (e.g., OpenFlow Switch or router), network link or a packet filter. The SR path is formed by the chain of these elements which are identified by a list of segment identifiers (SIDs). The SIDs can be defined globally with domain wide significance such that it is recognized by all network nodes or it can be defined locally within a node processing the packet [165].

SR meets well the requirements of scalability and flexibility of future 5G softwarized networks by minimizing the state information that should be stored in every network node along the delivery path because only the ingress/egress nodes are the ones responsible to maintain per-flow state information. It is worth noting that, the amount of forwarding rules in TCAM are reduced since packets are routed based on the list of segments and no need to maintain the path state or install flow rules in each switch or router along the delivery path. Thanks to the global view of the network provided by the SDN controller where the intelligence of the network can be programmed and be used to build segments and optimize them at any network topological change. For dynamic heterogeneous environments envisaged for future 5G softwarized multi-domain services and cloud applications, SR can provide a flexible network infrastructure that does not compromise

to network resources [153].

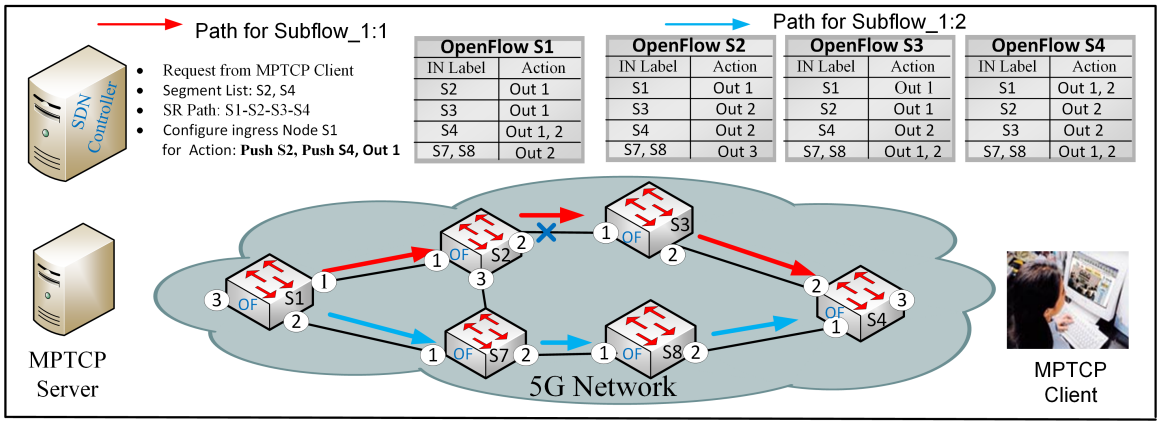

Figure 5.8: Segment routing operation in SDN-based 5G networks

Fig 5.8 illustrates an example of a SR operation in SDN-based virtualized 5G networks. In this work, a centralized control plane of SR approach is considered to implement the proposed idea. When an MPTCP client requests a new traffic from the MPTCP server, SDN controller calculates the transmission path P_{sf1}={S1→S2→S3→S4} based on the required constraints (e.g., delay and bandwidth of a link). This path is mapped as the segment label list and returned to the ingress switch S1 which then stores it in the packet header. The segment list consists of only the label that identifies the destination node in the SR domain such that, *SL*={S4}. The forwarding table of the ingress OpenFlow switch S1 is then configured by the SDN controller with this segment list labels. The switch will modify the packet by encoding the segments as MPLS label stacks to the packet header [165].

The packets of a subflow are then forwarded from S1 to S4 through intermediate nodes without any modification to the segment list. The top label is popped when the packets arrive at the intermediate nodes. The packets are then forwarded to the destination point using the segment node that represents the next label. SR can support and provide dynamic traffic recovery in multi-domain 5G SDN-based networks during network node or link failure [163]. For example, when link S2→S3 fails, then the SDN controller which continuously monitors the network topology can redirect the traffic on the backup path S1→S2→S7→S8→S4} with the associated *SL*={S2,S4}. At node S2, the top label is popped and the packet is transmitted to S7 via its Adjacency SID 3. At S7, the packet follows a path to its destination using its Adjacency SID at interface 2.

5.4.1 Mapping of Subflows Paths to SR Paths

The growth of traffic from video services (e.g., UHD 4K/8K video) drives future 5G networks to meet the requirements of supporting huge amount of traffic flows which are exchanged during peak hours. Consider a multimedia flow that is to be requested by the MPTCP client from the MPTCP server. Given the link capacity, l_c and the video bit rate b_r, the first task is to split a large flow into subflows and find multiple disjointed paths for the subflows. The second task is to map these subflows paths into SR paths and perform source routing based on QoS/QoE requirements. In order to achieve the first task, a MPTCP-based flow manager as described in the next subsection is introduced as a module running in the SDN controller. The mapping of subflow paths to SR paths is performed by finding the shortest list of SIDs that allows the packet to follow a given route, the detailed implementation will be covered in section 5.5.

5.5 QoE-aware MPTCP SDN-based SR Adaptation Framework

This section presents a QoE-aware MPTCP SDN-based SR adaptation frwamework for managing multimedia services in future network. The adaptation framework consists of the following modules: The MPTCP-flow manager, SR, QoE Management, Databases, Network information collector and the Configuration module which are integrated within the SDN controller.

5.5.1 SDN Controller

Fig 5.9 shows the function modules and interfaces of the proposed QoE-aware, MPTCP SDN-based SR framework. When the SDN controller receives a new request from the MPTCP client, it performs the path calculation through the MPTCP module and allocates the subflows to transmission paths. It then maps these subflow paths to SR paths as described in the previous subsection. The SR path of the new subflow is stored in the database. The SDN controller allocates and configures the SR path to the ingress switch node using the segment label list. The SDN controller communicates with switches in the data plane through the SouthBound Interface (SBI), using the OpenFlow protocol [154]. The OpenFlow protocol describes message exchanges that take place between the controller and an OpenFlow switch. video traffic tends to be bursty where the traffic

transmission has variable-bit-rate(VBR). A large burst of packets creates additional congestion and often leads to packet loss as well as reduced throughput. When a heavy flow arrives at the switch, the Packet-In Message is sent to the SDN controller. These messages are sent during state change or flow setup to reduce the CPU utilization and control traffic in OpenFlow switches and SDN controllers [156]. The Packet-Out Message enables the controller to manage the logical state of the switch, including its configuration and details of flows.

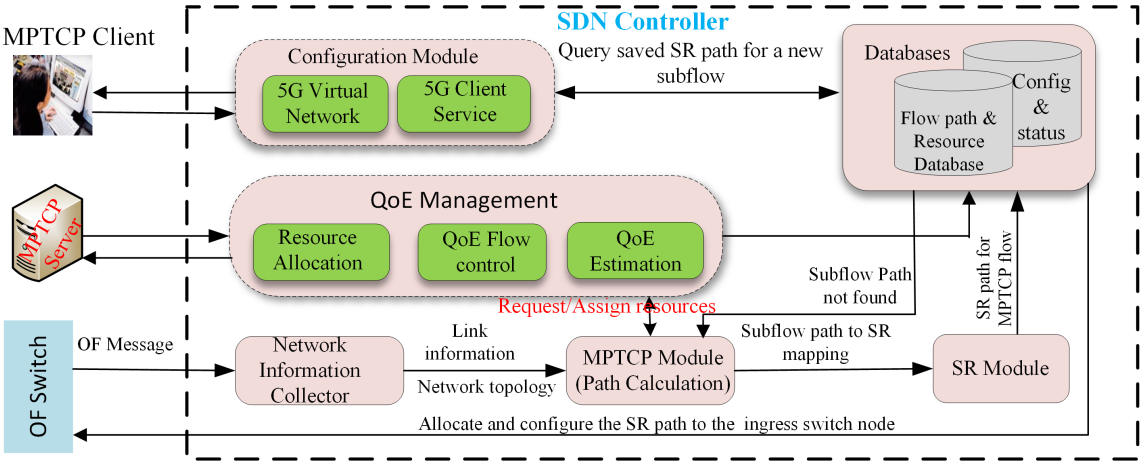

Figure 5.9: A QoE-aware, MPTCP SDN-based SR architecture

The controller maintains the database where all SR subflow paths are stored with their QoE requirements. When a new subflow of a MPTCP connection is uploaded to the SDN controller, the module queries in the database if there exists a path corresponding to this MPTCP connection. If the path exists, the subflow is allocated to a specific path of a previously assigned subflow's path of the same MPTCP connection, otherwise a new path computation for this subflow is performed using the MPTCP-flow manager/module. The new path is mapped to the SR paths and stored in the database so that it can be used later by subflows of the same MPTCP connection. The SDN controller architecture can provide network programmability capabilities which allows third parties (e.g., 5G virtual network operators) and other players in the QoE provisioning chain to setup their specific QoS/QoE control strategies.

5.5.2 MPTCP-Flow Manager

This module computes the shortest paths and then performs path allocations to the subflows (see Fig 5.9). In order to minimize the influences of link congestion to data transmission quality and the end-user's QoE, the MPTCP module employs admission

control for media delivery, such that, the flow is admissible only if on each link, the sum of the rates of the allocated subflows does not exceed the link capacity (l_c). In that case, the MPTCP module dynamically controls the number of generated subflows at the ingress source node. Based on the collected link information, the MPTCP module communicates with the QoE management module so that resources can be assigned to the calculated paths of subflows to meet their QoE requirements.

Instead of installing these subflow paths in SDN switches as forwarding rules, the proposed approach rely on SR approach where the forwarding table of the ingress switch is configured with an ordered list of segments [164]. The ingress switch then adds labels with an ordered list of segments to a packet header and forwards it to its destination point. This novel approach improves greatly the scalability, avoids link congestions and enables the deployment of effective multi-domain TE solutions in SDN-based 5G networks at a minimum cost.

5.5.3 Segment Routing Module

The computed subflow paths by the controller using the MPTCP-based flow manager have to be mapped to SR paths. The segment routing module is introduced to enhance greatly the SDN controller such that future 5G applications can exploit its features through the NBI. The SR assignment algorithm presented in [164] is considered to finds the minimal-length of SR paths that have to be mapped to the subflow paths. The SR path is the shortest list of SIDs that allows the packet to follow a given route. The list of SIDs consists of network nodes (e.g., node segments) or specific interface node (e.g., adjacency node). The source codes of this algorithm can be found at [164]. For example, given a multimedia flow f requested by an MPTCP client whose ingress node is OpenFlow switch S4 and the engress node is switch S1, then the complete path for subflow $sf1$ with intermediate nodes $\{S_2,..S_{N-1}\}$ is: P_{sf1}={S1→S2→S3→S4}. With reference to Fig 5.8, the algorithm takes this path and the graph of the topology as inputs. It maps the specific path of the subflow and returns the segment list as an output of the assigned SR paths. In this case, the ingress switch node S1 of the subflow path and the egress switch S4 are first considered. One of the assumption made is that, all links may have different cost. If there exists only one shortest path and it equals to the subflow path from S1 to S4, then the engress node is used as the SID node from S1 to S4 and the mapping of this subflow path to SR path ends

as shown in Fig 5.8. When a link failure occurs or it happens that the shortest path to be mapped is not equal to the subflow path or there exists more than one equal-cost shortest path, then other node segments can be considered as explained later in Chapter 6.

5.5.4 QoE-Management Module

Based on the specific QoE requirements for a service as defined at the network level QoS parameters, the QoE management module is to manage network resources dynamically for a certain multimedia flow application. The proposed solution can handle various sessions of multimedia applications through efficient QoE-centric control-based routing algorithms that redistribute the available network resources among the competing flows according to the end-users' QoE requirements.

5.6 QoE-Centric Multipath Routing Algorithm (QoMRA) over SDN

The QoE-Centric Multipath Routing Algorithm (QoMRA) is implemented on an SDN source routing platform using Mininet and POX controller. Although the OpenDaylight controller is more popular and has a very large codebase and many features to use, this work adopted POX controller because it is a Python-based platform that provides fast bridge learning and Mininet interactivity. The main objective of QoMRA is to enable service providers to route network traffic through explicit multiple disjointed bandwidth-satisfying paths that meet specific service QoE guarantees. Based on clients request, QoMRA should maintain an efficient use of network resources, reduce the potential for rejecting traffic-flow demands and network congestion. The traffic-flow demand in this work is defined as the request from the source $s \in V$ to destination $d \in V$ with requested TCP bandwidth B_w. In order to select k multiple shortest paths for the subflows from source to destination, the algorithm considers important information for video streaming such as the available link's bandwidth, delay, packet loss rate, jitter and link criticality. The link criticality parameter $c(l)$ is used for predicting the future traffic load of subflows on the link e. The $c(l)$ [165] is used to improve the routing performance based on the current load of each link e. The $c(l)$ is computed and updated by the SDN controller after every time interval t. The use of location of the network node is considered to define the

criticality of a certain link, the details can be found in [165].

The intent of $c(l)$ is to choose the links that can balance the loads across the network and avoid bottleneck links between pairs of communication nodes. The link criticality $c(l)$ can be defined using equation 5.1.

$$c(l) = \sum_{(s,d)\in TD} P_{sd}(c)/k \tag{5.1}$$

$P_{st}(c)/k$ is the occurrence rate of link e in the first k shortest paths whereas TD is the traffic demand that is recorded and stored in the database in every time t. A link is defined as a bottleneck when it provides a data rate that is lower than the required bitrate of a subflow to be transmitted through it. Table 5.1 shows the required TCP throughput for a specific video resolution which is selected by QoMRA when an MPTCP server receives a request from an MPTCP client. In case of buffering events due to congestion or link failure the algorithm can choose another routing path for a specific subflow as described in section 5.4. To mitigate the link failure or congestion, a link congestion index $k(i)$ is introduced, a function that increases sharply when the amount of traffic flow passing through the link e approaches its capacity. The idea behind $k(i)$ is to make use of paths with less link utilization during network congestion. When the link utilization grows beyond a certain threshold, QoMRA is able to capture the congestion state based on the information provided by the link utilization. However, QoMRA causes less overhead imposed by SR and MPTCP-flow manager modules during QoS parameters measurement that are to be used for shortest paths calculations. The overhead to the controller is reduced because only few intermediate nodes are used for data transfer from source to destination. When all the subflows paths are computed based on the link constraints and are assigned to SR paths, then an MPTCP connection can be established. The video player on an MPTCP client can then start to report the QoS/QoE metrics as the video streaming continues. The QoMRA algorithm consider first the network topology and then calculate the link delay, available bandwidth, packet loss and jitter for all paths in step 1. Based on these QoS metrics, the shortest paths are selected for network traffic transmission. Step 3 indicates that, if a path has been it is stored in the database so that when a another subflow with a bitrate less that the link capacity, similar to the previous subflow, then that path is returned and is used for transmission. The overall steps of QoE-centric multipath routing is summarized in algorithm 3.

Algorithm 3: A QoE-Centric Multipath Routing Algorithm (QoMRA)

input : flow f, number of subflows *sfi:j*, *N_topology*

1 *Compute link delay, bandwidth, packet loss, jitter for all paths*
2 *Find all subflow shortest paths* $p \in P$ *in the network based on Step 1*
3 **foreach** $p \in Psrc \rightarrow dst$ **do if** $p.used + sfi{:}j.b_r < l_c$ **then return** p
4 **else** Go to step 3
5 Map all subflow shortest paths p_{sf} into SR paths **output:** List of Segment labels *SL*
6 Save path p with its associated list of *SL* in DB_p
7 Initiate the MPTCP transmission of flows based on their QoE requirements
8 **if** f_{new} *is a new flow* **then** Query DB_p to find the paths for subflows of f_{new}
9 **if** pf_{new} *is not in* DB_p **then** Go to step 2 and issue path adding request for f_{new}
10 **end if**
11 **end if**
12 **end for**
13 Continue transmission as long as the l_c is greater or equal to the required TCP bandwidth
14 Use the $c(l)$ and $k(i)$ to avoid congestion

5.6.1 Video Clips/Sequence used for Experiments

The dynamic adaptive video streaming over HTTP service was applied to test the performance of the proposed algorithm and compared with the conventional TCP approach. DASH server was based on Apache and DASH client was based on VLC-DASH plugin [180]. For all experiments, the video sequence "Big Buck Bunny" shown in Fig. 5.10 with a resolution of 1920x1080 pixels of 9 minutes and 56 seconds. 5 representations of the selected video sequence with a minimum of 325Kb/s to a maximum of 5Mb/s were encoded using *ffmpeg* version 3.3.4 with the High Efficiency Video Coding (HEVC), and segmented based on GPAC MP4Box [53]. The video clips were then stored in the Apache Server. Such representations reflect the recommended and a typical video streaming rates used by YouTube. The video was further segmented into segments of sizes 2, 4, 6, 8 and 10 seconds.

Figure 5.10: Video clips/sequences used for implementations

Table 5.1: Required TCP throughput for QoE-centric algorithm

Video resolution	required TCP throughput (Mbps)
1080p	5
720p	2.5
480p	1.0
360p	0.725
240p	0.325

5.6.2 Experiemental Testbed based on SDN

The QoE-centric multipath routing algorithm was implemented in an SDN testbed based on Mininet and POX controller. In order to demonstrate the feasibility of our approach, two VMs installed with Linux (Ubuntu V16.04 LTS) and MPTCP v0.92 were used. The Mininet V2.2.2 was installed in one VM and used to model a scenario of a 5G datacenter network. As shown in Fig 5.11, the implemented topology consists of three levels with 8, 4 and 2 OpenFlow switches at the edge, aggregation and core layers, respectively. One MPTCP client is attached at each access switch. The network consists of redundant links at each level in order to accommodate subflows splitted by the MPTCP module. The POX controller running in the second VM is extended with an implementation of MPTCP-flow manager and the SR module. The SR module was implemented following a customization of source codes available at [164]. A video segment size of 6 seconds is

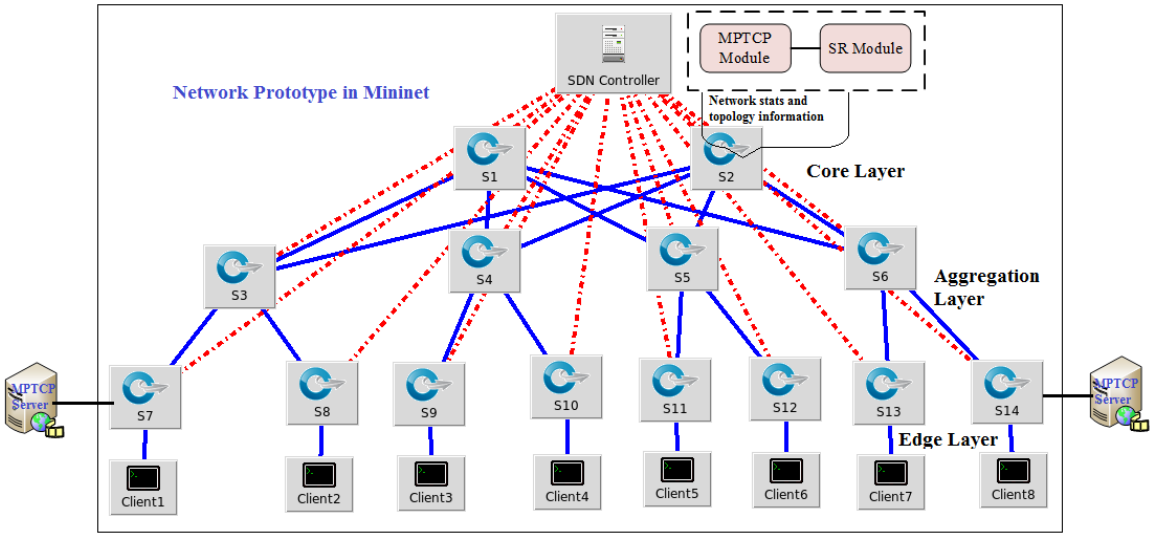

Figure 5.11: Experimental testbed

used in order to provide a good compromise between encoding efficiency and flexibility for stream adaption to bandwidth changes. The bandwidth and delay limits of (1Mb/s, 15μs), (2Mb/s, 20μs) and (3Mb/s, 30μs) are set for links in the edge, aggregation and core

layers respectively. A packet loss rate of (<3%) is configured at the core and aggregation links while the edge links are set free from packet loss. When the MPTCP is enabled, the default configurations of MPTCP V0.92 is retained and the MPTCP path manager is configured to a *full-mesh*. The aim is to create multiple subflows for each pair of IP addresses as described in section 5.2.1. This way, each MPTCP connection is limited to have only 3 MPTCP subflows. The video transmission is repeated 40 times from MPTCP client to MPTCP server. The proposed approach is compared with the regular TCP in terms of throughput, link utilization and the end-users' QoE.

5.7 Experimental Evaluation and Results

This section presents the results of the QoE-centric multipath routing algorithm described in section 5.6. The parameters (network throughput, link utilization, the impact of video segment length, quality switches, startup delays and success rate are considered to compare the performance of the proposed approach and the conventional TCP.

5.7.1 System Throughput

Throughput is the total size of successfully transferred data in a temporal time interval. It is calculated based on a **Payload_bits/download_time**; where a *Payload_bits* indicates the number of extracted bits of the video content per single unit time. Fig 5.12 shows the system throughput when an MPTCP client issues a video request from the server. It can be observed that, between 0-50sec, the regular TCP and our QoE-aware approach give almost similar performance. As the transmission of a video stream continues, the background traffic was introduced in the network using the *Iperf* tool. It is obvious that, the QoE-aware MPTCP SDN-based SR approach exploits multipath and segment routing features to increase the system throughput. Conversely, the regular TCP that uses single path for transmission performs poorly. It is also clear from Fig 5.12 that, the throughput of a regular TCP drops from 100sec. At this point, the user experiences re-buffering events due to throughput reduction problem while our approach maintains a good viewing experience to the user.

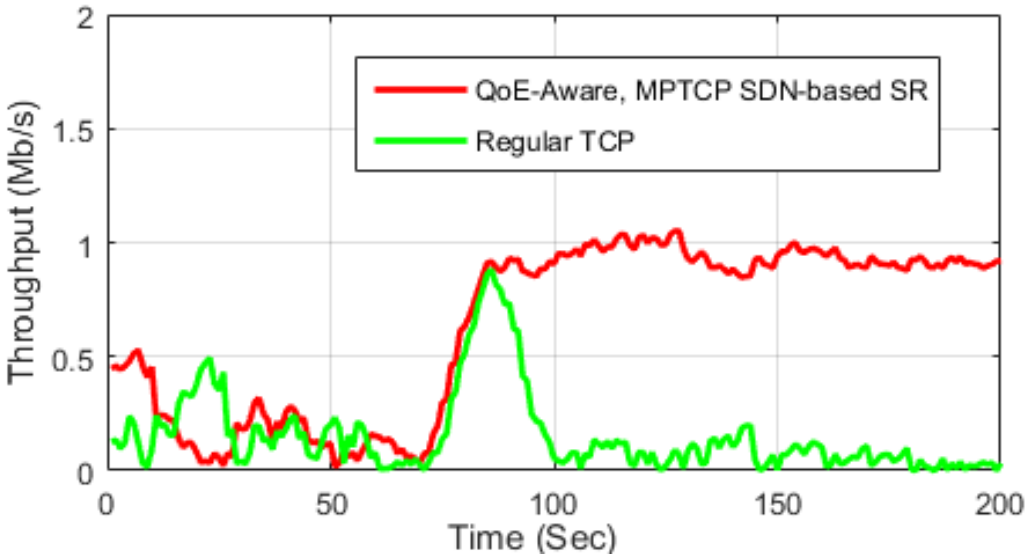

Figure 5.12: Comparison of throughput of QoE-aware MPTCP SDN-based SR and regular TCP

5.7.2 Link Utilization

The link utilization is computed from the amount of used bandwidth by a video content in a particular link over the total link capacity. Fig 5.13 compares the link utilization for the two approaches. Our QoE-aware proposal can maintain the link utilization above 50% for the whole duration of a video streaming due to multiple paths transmission using MPTCP and SR. As the transmission continues, the link utilization of a regular TCP starts to drop at 300sec. This is so because it uses a single path for data transmission such that when the requests increases, the load of OpenFlow switches also increases making TCP unable to react quickly to congestion in the link.

5.7.3 The Impact of Segment Length on Throughput

Segment length may have a decision on the performance of video streaming services and can affect the overall quality of the video delivered to users. For example, short video segment sizes can lead in poor performance of streaming services because of overhead that are produced by client's requests and the influence of network delay. In wireless networks, longer segment sizes may cause video stalls because of high bandwidth changes. This part evaluates the impact of segment size of adaptive streaming content and other factors such as HTTP server configuration. Fig 5.14 shows the impact of segment length of 2, 4, 6, 8 and 10sec on media throughput. Due to the influence of the network delay and the

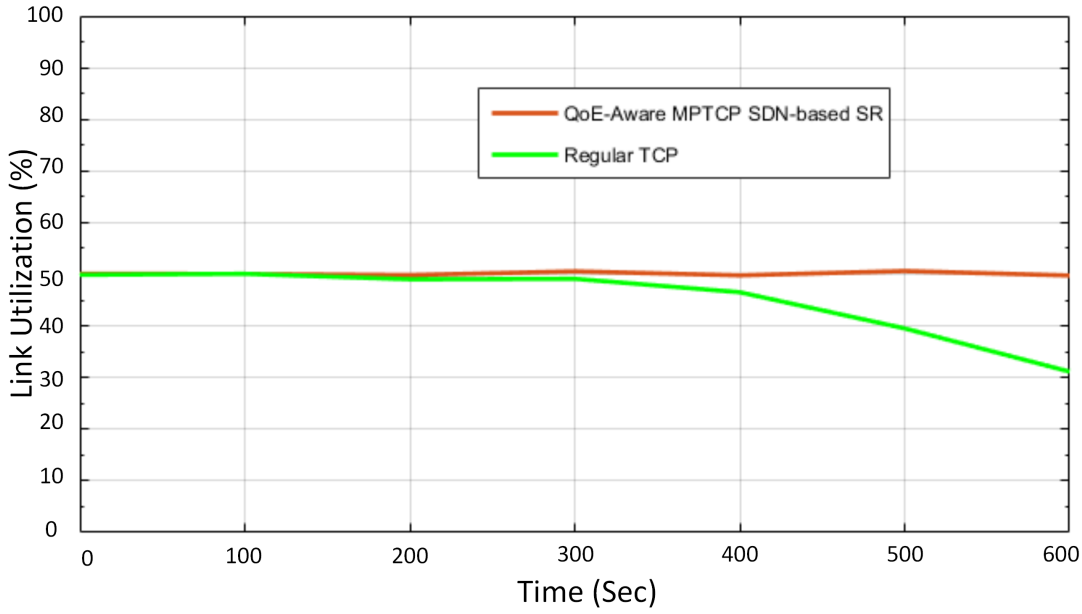

Figure 5.13: Comparison of link utilization of QoE-aware MPTCP SDN-based SR and regular TCP

overhead produced by the client requests, the regular TCP performs poorly compared to our QoE-aware MPTCP SDN-based SR approach. The throughput of a segment duration with 6 seconds shows an improved performance compared to that shown in Fig 5.12 because there is no background traffic imposed in this experiment. Shorter segments perform better in terms of throughput. However, when longer segments are used, the MPTCP client is unable to adjust quickly and flexibly resulting into poor performance. Although the bitrate adaptation process is enhanced for shorter segments and the buffer underflows is reduced but the server load is increased due to the fact that, segments requests are issued more often by the MPTCP clients.

5.7.4 Video Quality Switches and Startup Delays in Video Streaming

Video-quality switching and startup delays have an impact on the end-user's QoE in DASH. In order to characterize the impact of these parameters on the user's QoE, this section provide results of the evaluation using real-time experiments. The startup delays, number of quality switches and the success rate parameters are used to compare the two approaches. The success rate is defined as the percentage of segments at maximum bitrate over all segments used during a video streaming session. The quality switches define how many times the video quality changes from one bitrate to another during transmission due to change in bandwidth or delays in the networks. Three experiments were conducted

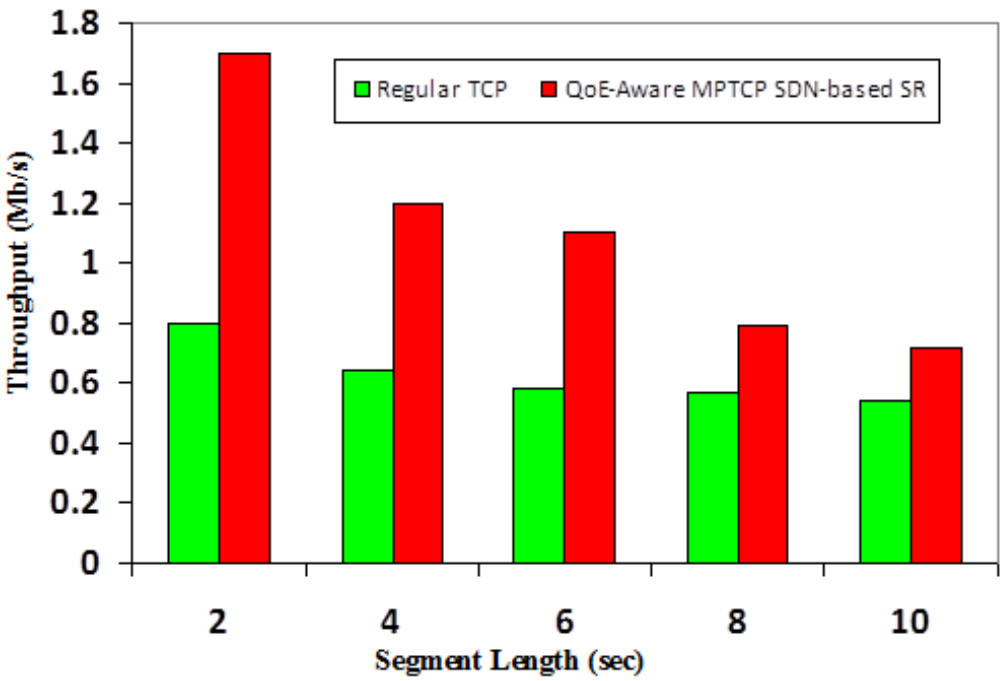

Figure 5.14: The impact of segment length on throughput

to investigate the effect of bitrate switching on video quality for QoE-aware MPTCP/SR and the regular TCP approaches. Figs. 5.15 shows the results of startup delays for three experiments using both the QoE-aware MPTCP/SR and the regular TCP approaches. The QoE-aware MPTCP/SR performs better with the average startup delays of 1.21sec, 1.97sec and 1.60sec for experiment 1, 2 and 3 respectively while the startup delays for regular TCP are 6.60sec, 6.30sec and 5.60sec for the same experiments. The QoE-aware MPTCP/SR results into an improved QoE to clients because of small number of startup delays during video streaming. This is not the case with regulat TCP with large number of startup delays that can annoy the user's watching experience during video streaming.

Fig. 5.16 shows the results of quality switching for three experiments. It can be clearly observed that, the proposed QoE-aware MPTCP/SR approach using SDN performs better in terms of quality switching compared to regular TCP. Fig. 5.16 indicate that, the switching impact for QoE-aware MPTCP/SR strategy is 2 while for regular TCP is 8. Small number of bitrate switching with minimum amplitude results into an improved QoE to the end-user while large number of switching may harm the video quality delivered to the user. However, even with large number of segment sizes, recent studies [181], [182] have shown that reliable bandwidth prediction and bounded bitrate guidance could boost the performance of the adaptation algorithms, especially when it is combined with rate

stabilization functions at the client side. An Autoregressive Integrated Moving Average (ARIMA) [183] using two time series models can forecast bandwidth and identify the optimal set of required bitrate levels for each client [184].

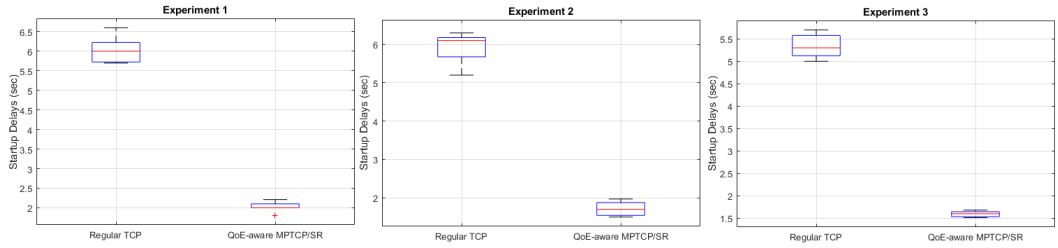

Figure 5.15: Startup delay for three experiments

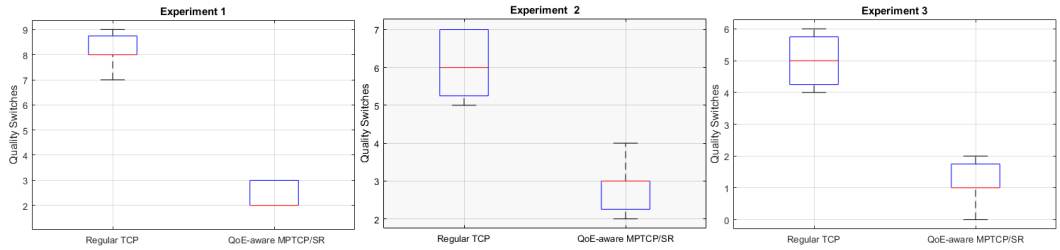

Figure 5.16: Quality switching for three experiments

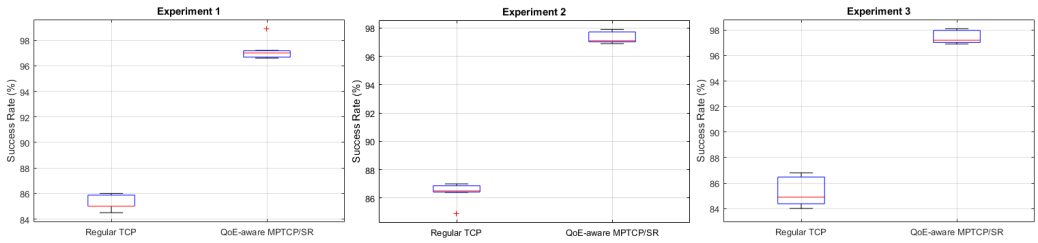

Figure 5.17: Success rate for three experiments

Fig. 5.17 shows the success rate of three experiments during the video streaming sessions. The proposed QoE-aware MPTCP/SR performs better compared to the regular TCP approach. This is so because SR and MPTCP are used to speed up the transfer of video flows from the client to the server when the request is made. This is not the case with regular TCP which only uses single path for transmission. For each video request, the QoE-aware MPTCP/SR fetches video segments at higher rate compared to regular TCP. As shown in Fig. 5.17, the success rate for experiment 1 is 97% for QoE-aware MPTCP/SR algorithm while the regular TCP archives the success rate of 85%.

Table 5.2 shows the summary of the average of startup delays, number of quality switches and the success rate from three experiments shown in Fig. 5.15, 5.16 and 5.17. The selected QoE results compares the performance of the regular TCP and the QoE-aware MPTCP/SR approach after performing a video request from the MPTCP server three times using a

6sec segment length. Low startup delays and few video quality switches provide a good user's watching experience with an improved video quality. It is clear from table 5.2 that, our QoE-aware MPTCP SDN-based SR proposal performs better compared to the regular TCP in all three experiments.

Table 5.2: Selected QoE comparison results for regular TCP and QoE-aware MPTCP SDN-based SR

	Regular TCP			QoE-Aware MPTCP/SR		
Experiment	Quality Switches	Startup Delays (sec)	Success Rate (%)	Quality Switches	Startup Delays (sec)	Success Rate (%)
1	8	6.60	85	2	2.21	96.64
2	7	6.30	87	3	1.97	97.80
3	5	5.60	84	1	1.60	97.01

5.8 Summary

It is evident that, efficient transfer of large flows, especially multimedia traffic is crucial to the performance of 5G multi-rooted topology networks where multiple paths exist between pair of hosts. In an attempt to address the challenges of transmitting and delivering high-demanding multimedia applications such as UHD videos with high QoE in 5G SDN-based networks, this chapter has presented a QoE-centric multipath routing mechanism that dynamically forwards the number of subflows using SDN controller and perform source routing to these subflows using SR paradigms. The presented QoE-aware MPTCP/SR algorithm indicates that, a joint use of MPTCP and SR can improve the performance of future 5G network softwarized. Compared to the conventional TCP approach, the proposed QoE-aware MPTCP SDN-based SR shows better performance in terms of increased network throughput, link utilization and the end-users QoE. An integration of SDN and NFV for QoE control and management of multimedia services will be presented in Chapter 6.

Chapter 6

QoE Management of Video Streaming Services using MPTCP and SR in Softwarized Networks

As multimedia services/applications continue to grow, ISPs have adopted SDN/NFV to implement TE in order to improve the network efficiency, application performance and the end-users QoE, notable examples include Google's B4 SDN/TE [185] and Microsoft SWAN [186]. Chapter 5 demonstrated that, MPTCP and SR implementations can be integrated by cutting-edge technologies such as SDN in order to achieve load balancing, reliable communication and better network resources utilization that leads to higher network throughput and the end-user's QoE [161], [167]. However, Chapter 5 did not consider the integration of SDN and NFV to improve the end-users' video quality.

This chapter provides original practical TE solutions based on MPTCP and SR leveraging both SDN and NFV paradigms. The main objective is to facilitate routing and speed up the transfer of large amount of multimedia applications between source and destination points. To improve the video quality, the use of multiple shortest paths for MPTCP subflows transmission is employed. The Multi-flow commodity and constrained Shortest Path Model (MSPM) [187] is used to choose important intermediate nodes to perform source routing using SR paradigm. This chapter extends Chapter 5 where it was demonstrated that utilizing MPTCP/SR in SDN-based networks can improve system performance for video streaming services. The aim is to meet future multimedia network bandwidth aggregations and transfer large amount of data through multiple disjointed paths as computed by the SDN controller. This novel concept is extended to further consider the NFV implementation in the thesis. The contributions of this chapter are 3-fold: a novel QoE- Softwarization concepts using the integration of SDN and NFV is given

along with a QoE-aware MPTCP SDN/NFV SR-based architecture that provides an efficient orchestration, QoE control and management of video streaming services. in future softwarized infrastructures. The ETSI MANO (OSM) framework is employed for the management and orchestration of resources. The chapter further presents a system model and a QoE-based multipath source routing algorithm called **"QoEMuSoRo"** that forward traffic using SR paradigms over softwarized systems. In addition, Path protection and Recovery of Link Failure (PathReLieF) algorithms are proposed to achieve dynamic route selection and link/node failure recovery over softwarized networks.

The rest of this chapter is organized as follows: Section 6.1 provides related work while section 6.2 provides novel concept called "QoE-softwarization" in SDN and NFV. Section 6.3 presents the considerations of MPTCP over softwarized networks. Section 6.4 provides the description of path protection and traffic recovery mechanisms with SR in softwarized networks. Section 6.5 provides the Service Function Chaining (SFC) strategies using SR in future softwarized network while section 6.6 provides a novel QoE-aware MPTCP and SR-based approach over softwarized infrastructures leveraging the integration of SDN and NFV. Based on MCSPM and the concept from graph theory, section 6.7 presents a formulation of a system model and the proposed QoE-based multipath source routing algorithm. Section 6.8 presents the performance and evaluation of the proposed SDN/NFV system. Finally, section 6.9 summarizes this chapter.

6.1 Related Work

This section presents first related works that consider three categories of research fields regarding video streaming. The first works use MPTCP without SDN while the second category employs MPTCP over SDN. The third category are those works that either use SR to perform service function chaining or traffic recovery and link/node failure mechanisms in SDN. It is important to note that, the related works presented in section 5.1 neither consider path protection nor link/node recovery mechanisms and they are therefore related to chapter Five only.

In an attempt to improve the video quality using MPTCP, James *et.al* [188], perform video streaming over DASH to investigate the benefits of MPTCP to user's QoE under ample and stable bandwidth. Wu *et.al* propose a quAlity-Driven MultIpath TCP (ADMIT) [161] strategy that improves the video delivery quality over multiple paths by integrating the

utility maximization based on Forward Error Correction (FEC) coding and rate allocation. In order to schedule the delivery of video packets and increase the QoE of viewers, the concept of MPTCP-based cross-layer scheduler is introduced in [162] that leverages the information from both application and transport layers. However, these studies neither use SDN nor NFV for improving the video quality and therefore lack the global control feature offered by softwarized infrastructures which are considered to be potential technologies in future 5G networks. They do not also consider how the network services can be implemented by a set of VNFs in future SDN/NFV-based networks [167], [189].

Kukreja *et.al* [178] propose an Openflow-based SDN controller and implement a MPTCP path manager where switches are configured to forward subflows over different paths. The SubFlow Optimizer (SFO) is presented by Joshi *et.al* [190] to enable load balancing by providing MPTCP hosts an optimum number of subflows and assign the best paths to new established subflows in SDN environment. Using the Linux's Kernel-based Virtual Machine (KVM), the behavior and performance of a virtualized proxy as an instance of NFV is analyzed in MPTCP connections over cellular network by Chung *et.al* in [191]. The multipath routing problem in NFV with regard to practical factors of both NFV characteristics and multipath routing is formulated in [192]. A Multi-flow update problem is formulated as a mixed integer programming in [167] to reduce the update time of flows in Software Defined NFV (SDNFV) systems. Note that, none of these MPTCP-based SDN/NFV solutions consider to improve the end-user's QoE using real-time video communication. In addition, as mentioned earlier in chapter Five, MPTCP implementation in SDN/NFV would result into installing large number of flow rules in switches leading to overhead at the controller and increased deployment cost.

As shown in section 5.4, SR is an appealing solution for reducing the number of forwarding rules taking the fact that each SDN switch on the path needs to have an entry for a traffic-demand to forward its packets to the next hop. SR provides capabilities to reduce the number of forwarding rules by encoding TE paths into SR paths. SR in [179] is reported to save up to 88% forwarding rules overhead and handled more than 92% flow updates in real-time compared to traditional solutions. A recent SDN [193] SR-approach, indicates an approximately 50% reduction in the number of control messages between source and destination nodes. Practical TE through shortest paths in SDN WANs using SR is employed by Trimponias *et.al* [194] where only few intermediate nodes are selected to route all traffic in the network. A network model based on IPv6 Segment Routing

(SRv6) [189] is used to steer traffic within a Linux-based NFV host that supports large number of VNFs in the NFV architecture. Motivated by the capabilities of SR to reduce flow rules, this chapter introduces a novel QoE-aware MPTCP-enabled SR-based strategy to control and manage video streaming services in future softwarized networks leveraging the integration of SDN and NFV. The aim is to provide an efficient orchestration, QoE control and management of video services while ensuring path protection and traffic recovery mechanisms when link/node failure occurs in the softwarized network. Section 6.2 provides a novel QoE-softwarization in future SDN and NFV networks.

6.2 QoE- Softwarization in Future Networks using SDN and NFV

As the first step in this research project into the design and development of QoE control and management approaches for multimedia services, a novel QoE aware SDN-enabled, NFV-based architecture along with the concept of *QoE-Softwarization*, (see [63]). The QoE-Softwarized architecture is set to enable mobile operators and service providers to deliver quality services through an autonomic software lifecycle management approach in 5G. The proposed architecture is mainly designed to provide flexible networking and configurable approach for QoE provisioning through the integration of SDN and NFV. This approach takes the softwarization advantages by adopting the principles of SDN and NFV in order to meet the Key Performance Indicators (KPI) of 5G. It also complies with SDN-NFV requirements which translate to 5G KPRs such as consistent service availability, network reliability and survivability, accessibility/retainability, throughput and minimal latency [63].

Since NFV is also integrated in the proposed architecture, Virtual machines (VMs) which are able to obtain different requirements for network resource allocation and utilization can be deployed in any location within the QoE-Softwarized architecture. The VMs may have the required quality measures/parameters of QoS and QoE of a particular multimedia application specifically in terms of network bandwidth, scheduling latency, jitter and stalling etc. Based on the decoupling of services from resources offered by NFV and SDN paradigm by detaching lifecycle management from physical constraints, the locations of Virtual Network Function (VNF) may change in the network. For example, during malfunctioning of any VM, the affected VNFs can be migrated to another location in order to avoid service delivery interruptions. A QoE-aware SDN controller shown in

Figure 6.1 is proposed to provide connectivity and allocate paths dynamically among VNFs to achieve QoE for VNF services. Such an SDN controller acts as an enabler of chaining NFVs. In fact, it provides an ability to realize the multimedia network services by dynamically chaining together VNFs and direct traffic flows into those VNFs chains.

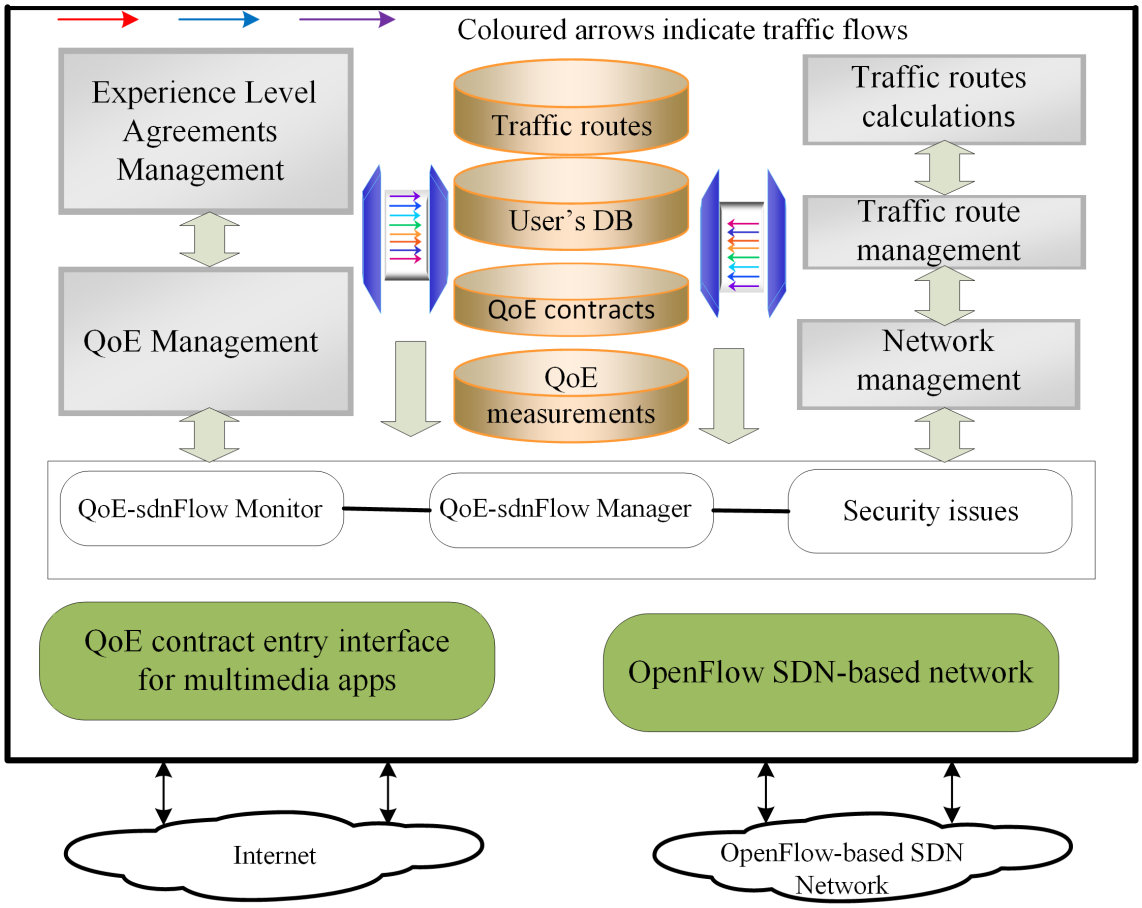

Figure 6.1: SDN controller with QoE entities

In the proposed SDN controller, the QoE-sdnFlow Manager performs the overall customer experience, QoE optimization and QoE-aware 5G network management as well as enhancing the quality of business through a well-defined QoE contracts between service providers, operators and end users. The *QoE-sdnFlow Manager* is also set to acquire network topology information and implement QoE based network policies and techniques by using different control algorithms for QoE traffic predictions, admission control, radio resource allocation, load balancing and user density prediction. It can also perform and activate network-wide operations related to multimedia streaming, packet prioritization and caching node capabilities for some multimedia contents or applications. The *QoE-sdnFlow Monitor* performs the QoE estimation and QoE measurements per multimedia traffic flow, selection of the time period for data acquisition and generation of QoE based inputs and reporting this to

the *QoE-sdnFlow Manager*. Furthermore, this entity/module performs multimedia traffic classification in order to know the type of traffic using statistical/parametric analysis.

6.2.1 Application Scenario and Use Cases

Figure 6.2 shows an application scenario based on the proposed architecture where a user requests a multimedia application from a service provider. In this scenario, various VNFs are able to provide a support to intensive applications (e.g., live video conferencing and IPTV in home networks) that demand better quality and require high bandwidth.

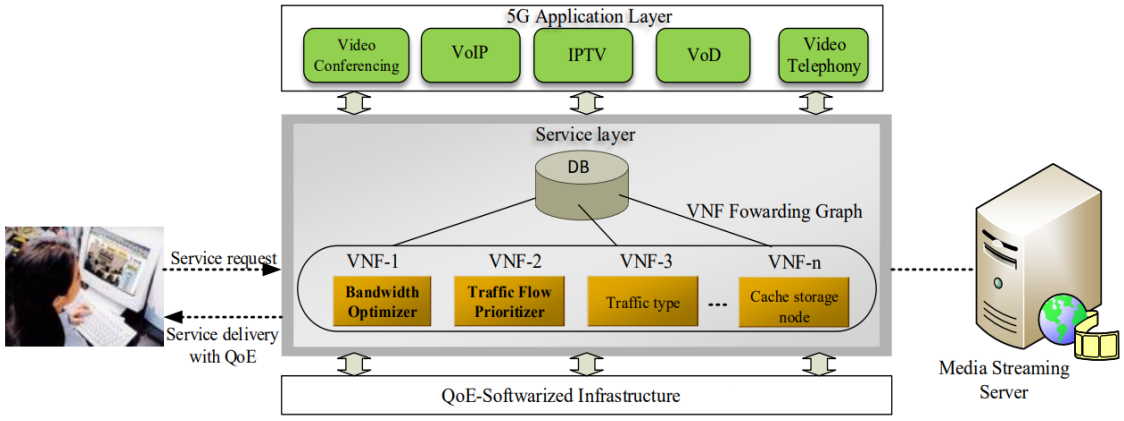

Figure 6.2: QoE-driven video streaming over softwarized infrastructure

The service layer consists of a range of VNFs. Depending on the type of multimedia application (e.g., IPTV, VoIP, live video conferencing) requested by the user, the network service functions provide a set of functional operations to the user and fulfill his/her personalized QoE application requirements. For the case of live conferencing, the QoE-Softwarized is able to provide sufficient QoE through multi-point to multi-point connectivity among conference attendees using a set of VNFs. For every VNF, a service graph is formulated which establishes the VNFs Forwarding Graph stored in the VNFs database (DB) based on the request from the end-users. The QoE-sdnFlow Manager or the SDN-NFV orchestrator instantiates the DB as soon as the user requests a multimedia application with specific QoS and QoE indices. It then responds based on feedback obtained from the QoE-sdnFlow Monitor in order to maintain/optimise a user's end-to-end QoE [63].

6.3 MPTCP Implementations over Software Defined and Virtualized Networks

SDN and NFV have emerged as a cost efficient solution to promote the delivery of an E2E services and network infrastructure capacity scaling in supporting the huge demand of emerging multimedia services. The integration of SDN and NFV looks to be an appealing solution for efficient orchestration and management of video streaming services such as 2D/3D, 4/8K broadcast content in future softwarized networks. By properly chaining VNFs, different services can be distributed over the NFVI forming a process refereed to as *Service Function Chaining (SFC)* [81]. SFC is an instantiation of an ordered set of network service functions and the subsequent steering of traffic through those functions. Similar to overhead resulted through MPTCP implementations in SDN, the suboptimal placement of VNFs in NFV service chain increases the flow rules in the switches and can also affect the efficiency of bandwidth utilization [195]. However, by exploiting an implementation of SFC as shown in section 6.5, the VNF chain can be represented in a SR packet Header (SRH) containing a SR path as an ordered list of segments. As shown in Fig. 6.3, TCP connections called subflows, each having its own sequence number, are established under the MPTCP layer as described in the previous section 5.3. These subflows can be transmitted by multiple paths simultaneously using a flow distribution function of MPTCP. Fig. 6.3 shows a set of *VNFs*={VNF_1....,VNF_n} that can be arranged together to form a SFCs, for example regarding video optimization, network acceleration, HTTP header enhancement and load balancing. SDN can be used for dynamic orchestration and communications between SFCs, the network apps/services in the cloud and user's mobile terminal. In such a case, SDN controller can be used to allocate paths and provide connectivity dynamically among VNFs and therefore acting as an enabler of NFVs. It is worth mentioning that, SDN provides an ability to realize the video streaming and other network services by dynamically chaining together VNFs and direct traffic flows into those VNFs chains. Using network service chaining as shown in Fig. 6.3, the OTT service providers can apply QoS/QoE and user prioritization policies to network service chain resource management that involves users connected at different network points, in a distributed manner [196].

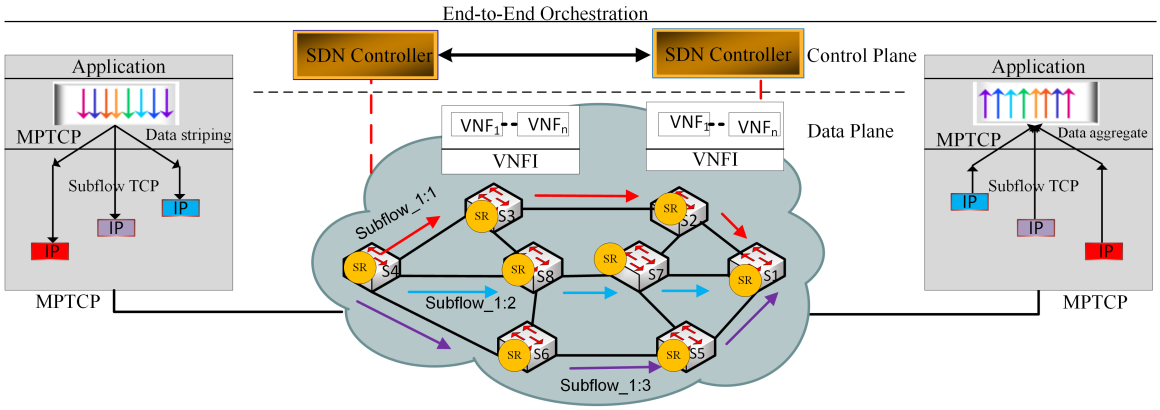

Figure 6.3: MPTCP implementations in softwarized infrastructures

6.4 Path Protection and Traffic Recovery Mechanisms with Segment Routing in Softwarized Networks

This subsection presents path protection and traffic recovery mechanisms due to link/node failures in softwarized networks using SR technology. Chapter Five provided the introduction to segment routing in SDN where the SIDs can form a chain consisting of a set of network elements where packets are routed within a softwarized infrastructure. The main three categories of SIDs are:

- **Node SIDs**: A node segment with domain wide significance that forwards MPTCP subflows on the shortest path towards the destination node (e.g., node segment labels 100 and 102 in Fig 6.4 are the Node SIDs).
- **Adjacency SIDs**: A specific node interface with local significance that forwards MPTCP subflows over the corresponding adjacency. With reference to Fig 6.4, label 10A and 10B indicate Adjacency SIDs for Link A and Link B towards S7 and S8 respectively.
- **Service SIDs**: A service segment assigned to each node having a local significance to represent each service a node provides to the network. Effective service chaining along TE paths in SDN/NFV networks can be easily enforced using *Service SIDs* as shown later in this work.

As shown in Fig 6.4, when the request from MPTCP client arrives at the MPTCP server, SDN controller computes the shortest paths for the subflows. For clarity, only two subflows are considered as indicated in the figure. In this case, p_1={S1→S2→S3→S4} and

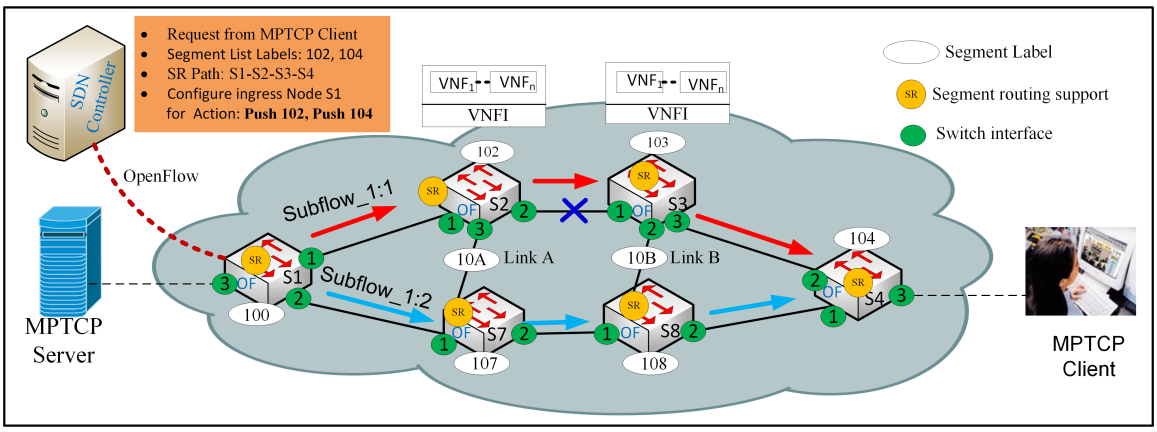

Figure 6.4: An example of SR Operation in softwarized networks

p_2={S1→S7→S8→S4}. *sf1:1* can reach at S4 via S2 by using *Node SIDs* with segment list, SL={102,104} as encoded by the SDN controller at switch S1 in the SR header. On arriving at S2, the packets will be forwarded to S3 based on the active top label segment of SID 102. The top label defines an active segment that instructs packet forwarding to the next segment/label along the delivery path. At S3, the MPTCP subflow packets are then forwarded to its endpoint, S4. Arguably, different from the steering packets using the Resource reservation Protocol-TE (RSVP-TE), it is important to note the flexibility and the benefits of SR for routing packets without storing routing information at S3.

Path protection and dynamic network recovery from failure is an important aspect of QoE control and management in SDN/NFV networks. This is so because network failures (due to a failed link or node) directly affect transmission quality and ability of service providers to meet their Experience Level Agreements (ELAs)[1]. Typical failures in SDN/NFV network include: *Link*, *Node* and *Shared Risk Link Group (SRLG) failures* where multiple links can fail together. When network failures occur in the network, segments can be repaired such that new backup paths are computed for each link in the primary path. The scope of this chapter is mainly on link failures using SR.

With reference to Fig 6.4, when link S2→S3 fails, the topology is updated by the SDN controller which has the global knowledge of the network. In such case, link S2→S3 is pruned before new backup paths are computed. A failover table is created where *backup actions* for each source-destination points are also indicated. The controller encodes the required segment list on the ingress-node following the new computed backup paths. As an example, Fig 6.5 shows the primary tables and the failover table for S2 when link

[1]Experience Level Agreements (ELAs): Indicates a QoE-enabled counterpiece to traditional QoS-based Service Level Agreements (SLA) that conveys the performance of the service in terms of QoE. ELAs establish a common understanding of an end-user's experience on the quality levels whiling using the service.

S2→S3 fails. It is important to note that, the FT_1 corresponds to failure of link S1→S2 at interface 1 of S2, FT_2 for S2→S3 at interface 2 of S2 and FT_3 for S2→S7 at interface 3 of S2. For simplicity only Failover tables for S2 in Fig 6.5 are illustrated.

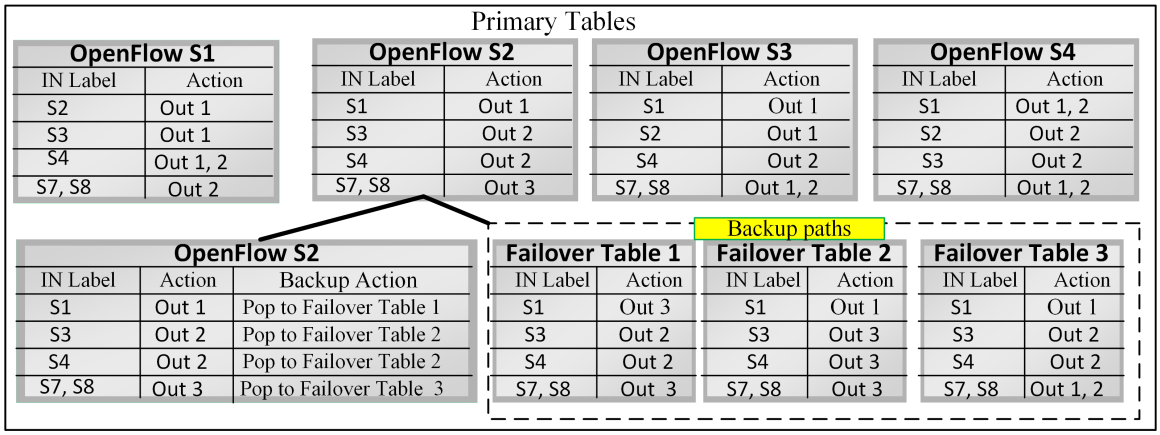

Figure 6.5: Path protection using backup actions configured in failover tables (FT).

Consider the subflow *sf1:1* traversing through p_1={S1→S2→S3→S4} encoded with segment list, SL={102,104}. When link S2→S3 fails, then label 104 is popped and S2 is redirected to **Failover Table 2** with the backup path Bp_1={S2→S7→S8→S4}. This path is encoded with SL={104} where normal forwarding is performed towards S4. An important aspect to improve the end-user's QoE is to ensure that all MPTCP subflows are protected against link or node failures in the SDN/NFV network. This work proposes to prune the links which are directly connected to a failed interface during the initialization of a failover table. For example, when S2 detects a failure at interface 2, the SDN controller updates the topology without link S2→S3. In case of a node failure, then all bidirectional links to that node are pruned. As an example, link S1→S2, S2→S3 and S2→S7 will be removed from the network topology when S2 fails. In such a scenario, then the number of shortest paths for MPTCP subflows are reduced which may affect the performance of SR approach. However, node failures rarely happen in IP networks. In that respect, an attention is given to path protection mechanisms based on link/node failures as shown in Fig. 6.4.

6.5 Service Chaining for OTT Service Providers using SR in Softwarized Networks

As stated in [189], SFC is an ordered list of abstract service functions that should be applied to a packet. Typical examples of these service functions or VNFs include: Deep

Packet Inspection (DPI), Server load balancers, HTTP Header Enrichment (HHE), NAT44, NAT64, TCP optimizers and Firewalls. The set of switches/nodes where the VNFs can be instantiated is called the NFV Infrastructure. As shown in Fig 6.4, the NFV nodes are SR-enabled switches which can also run VNFs. While NFV promises to provide better flexibility solutions in future networks (e.g., 5G), SFC eliminates greatly the constraint associated with today's operators who are suffering from large management overhead on implementing their network functions. With the current use of IPv4 and IPv6 in the SDN/NFV data-plane, each VNF IP address can correspond to a particular SID. The VNF chain can therefore be represented in the SRH which contains the SR path as an ordered list of service segments. Fig 6.6 shows NFs that are served for different enterprise users. According to service requirements, one NFs can be selected by one or more users from different enterprises. With three service chains, each chain consists of a series of ordered functions which can be represented in SR by their Service IDentifiers. The operator

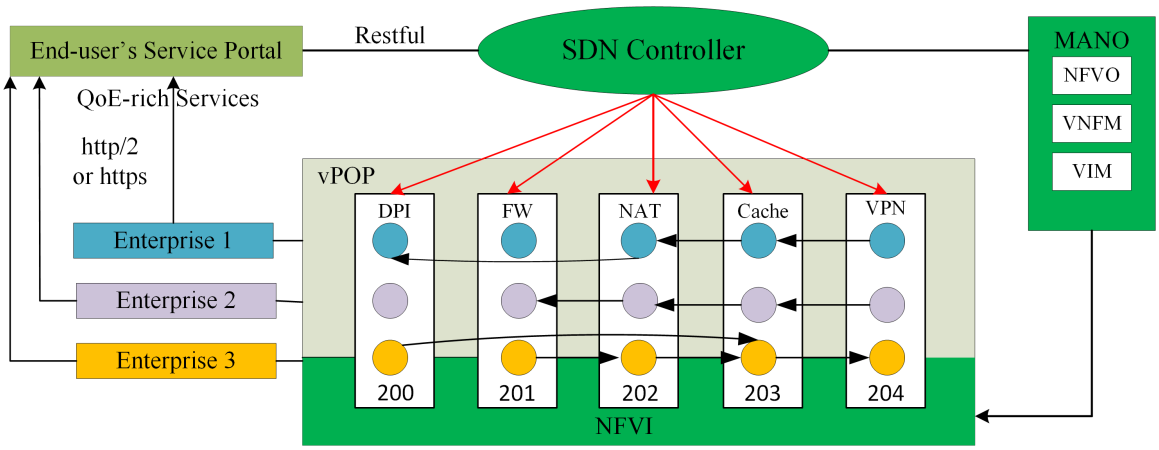

Figure 6.6: A scenario of service function chaining for enterprises using SR

managing the network could assign the SIDs as follows: 200, 201, 202, 203, 204 for DPI, Firewall, NAT, Cache and VPN respectively. Enterprise 1 that pays attention to intranet service, needs a DPI, Cache, NAT and VPN functions and the operator would push the SR header with segment list SL_{SFC1}={200, 202, 203, 204}. Enterprise 2 that pays attention to intranet service, needs a Firewall, NAT, Cache and VPN functions. The operator would push the SR header with SL_{SFC2}={201, 202, 203, 204} and forward the packet. For Enterprise 3 who needs all services, the operator would push the SR header list with segment list, SL_{SFC3}={200, 201, 202, 203, 204} and forward the packet.

6.6 The QoE-Control and Management Architecture of Video Streaming Services in Softwarized Networks

In order to meet the aspects of QoE control and management of multimedia services in future softwarized networks, this section presents a novel QoE-aware MPTCP/SR that integrates SDN and NFV architecture as shown in Fig 6.7. The proposed architecture can achieve an optimized E2E QoE level for the end-users in future softwarized networks. Video flows are transfered through multiple disjointed shortest paths for MPTCP subflows and the controller introduced in section 5.5 can perform source routing using SR paradigm. More importantly, it can learn and adapt to changing network conditions or media contents as well as overcoming link/node failures. The proposed SDN/NFV system consists of the Data plane and the QoE-control and management plane.

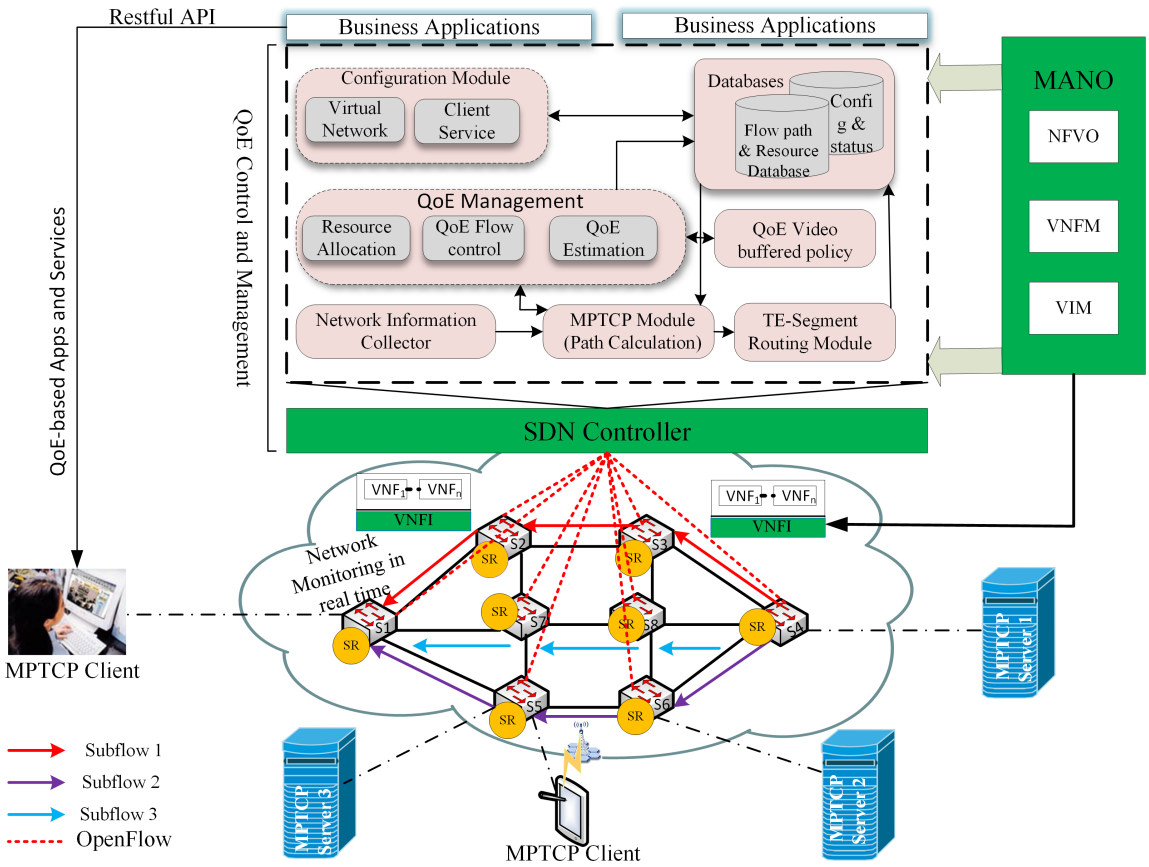

Figure 6.7: The proposed QoE-aware, MPTCP/SR-based in softwarized networks.

6.6.1 Data Plane

The data plane consists of SDN switches that support SR technology. It forms the data acquisition layer which consists of a set of different Virtual Network Functions (VNFs) arranged together depending on a set of network traffic flows. It also represents the forwarding layer of the network where many software/hardware devices (e.g.,virtual routers and virtual switches) are interconnected using virtual or wired connections or common wireless radio channels.

6.6.2 QoE Control and Management Plane (QoCoMa)

The POX controller was extended with three functional modules, namely, the MPTCP-based flow manager module, the TE-Segment Routing module and the QoE management module. POX was used in this implementation because it provides a Python-based platform that offers a fast bridge learning and Mininet interactivity easily. The QoCoMa plane consists of other entities such as: Network Information collector and the Database module.

6.6.3 Network Information Collector (NIC)

This module performs collection of network information and QoE requirements of end-users, system events and any network topological changes (e.g., during link or node failure).

6.6.4 MPTCP Module

This module computes the shortest paths and then performs path allocations to MPTCP subflows as described in section 5.5.2. Based on the collected link information, the MPTCP module communicates with the QoE management module so that resources can be assigned to the calculated paths of subflows to meet their QoE requirements. Instead of installing these subflow paths in SDN switches as forwarding rules, SR approach is used where the forwarding table of the ingress switch is configured with an ordered list of segments [164]. The ingress switch then adds labels with an ordered list of segments to a packet header and forwards it to its destination point.

6.6.5 TE-Segment Routing Module

As implemented in section 5.5.3, the TE-SR module maps the computed subflow paths to SR paths. With reference to Fig. 6.7, the proposed algorithm running over softwarized networks takes this path and the graph of the topology as inputs. It maps the specific path of the subflow and returns the segment list as an output of the assigned SR paths. More details of SR implementations are available in section 5.5.3.

Fig. 6.4) shows the path protection and link recovery using *Backup Actions* configured in *Failover Tables* when a link/node failure occurs in the network, then path protection mechanisms discussed in the previous subsection are considered (see Fig 6.5).

6.6.6 QoE-Management Module for Softwarized Networks

The QoE management entity presented in section 5.5.4 is extended to perform various network management and ensure that the overall KPIs of the softwarized networks are fulfilled. It also ensures that proper resource allocations, QoE-centric control and QoE estimation related to video streaming services are achieved. It further provides softwarized infrastructure domain monitoring and multimedia service availability in order to enhance the QoE of the end users. In order to ensure the overall system performance of a multimedia service, the QoE management module takes measurements of KPIs related to QoE metrics such as stalling, video quality and network metrics such as packet latency, jitter, network throughput and packet loss. It also performs autonomic fault detection and recovery mechanisms for ensuring that any point of failure in the softwarized systems can be configured and recovered without affecting the end-user's QoE service delivery chain. The MANO is responsible for managing and orchestrating VNFs. Through the NIC module, any changes of users (e.g., joining or living the connection) and the running status of the SDN/NFV system can be reported immediately to the SDN controller.

6.6.7 NFV Management and Orchestration (NFV MANO)

According to the ETSI [82], network functions virtualization management and orchestration (MANO) is responsible for managing and orchestrating virtualized network functions (VNFs). It consists of the NFV Orchestrator (NFVO), and the Virtual Infractructure Manager (VIM). The NFVO performs orchestration and lifecycle management of physical and

software resources that support the infrastructure virtualization. The NFVO also performs global resource management, network services (vCompute, vStorage and vNetwork resources) within one network operator's infrastructure domain. The proposed QoE-aware MPTCP SDN/NFV SR-based system can provide an efficient orchestration, QoE control and management of future multimedia services, service instantiation, validation and authorization of NFVI resources requests. The VNFM is responsible for lifecycle management of VNF instances, overall coordination and adaptation role for configuration and event reporting between NFVI. A single or multiple VNF instances of the same or different types can be managed by a VNFM. The VIM controls and manages NFVI physical and virtual resources. For the purpose of implementation, this work adopts the Open Source MANO (OSM) [197] to instantiate the VNFs in softwarized networks.

6.6.8 Configuration Module

This module provides interfaces for virtual and physical network resources setup. It also provides interfaces for end-users service QoE configurations (e.g., QoS/QoE-based flow transmission rule for multimedia applications such as throughput, packet loss). The configurations of these resources (for virtual and physical) and their associated QoS/QoE policies are stored in the configuration and resource databases as shown in Fig. 6.7.

6.6.9 Database Storage Module

The configuration parameters, monitoring status reports and different fine-grained resources are stored in the database module. It also maintains all SR subflow paths with their QoE requirements. When a new subflow of a MPTCP connection is uploaded to the SDN controller, the module queries in the database to look for an existing path that corresponds to this MPTCP connection. If the path exists, the subflow is allocated to a specific path of a previously assigned subflow's path of the same MPTCP connection, otherwise a new path computation for this subflow is performed using the MPTCP-flow manager/module. The new path is mapped to the SR path and stored in the database so that it can be used later by subflows of the same MPTCP connection.

6.6.10 The SDN Controller

The SDN controller is implemented as an extension to POX controller. Computation of shortest paths of subflows is performed using the MPTCP module when the request is uploaded to the controller as described in section 6.6.4. The controller allocates the subflows to transmission paths using the MPTCP module and then maps these subflow paths to SR paths using the TE-SR module (see in Fig 6.7).

6.7 System Model and QoE-based Multipath Source Routing Algorithm in Softwarized Networks

A softwarized network is considered which is represented as a directed graph G = (V,E) where V is the set of nodes (switches or VNFs) and E is the set of edges. Each $e \in E$ is associated with a non-negative integer link weight denoted by $W(e)$. In real world network scenarios, the link weights indicate a non-fixed parameters such as link bandwidth, packet loss, delay or link utilization. Such information is used to compute the optimal multiple set of h shortest paths from source to destination. Available link bandwidth denoted b_w is set to connect a pair of softwarized nodes in the network and B^k_{sf} is used as the required bandwidth of a subflow sf of an MPTCP connection k. The shortest path of a traffic-flow demand f from the source $s \in V$ to destination $t \in V$ is denoted by $p(s, t)$. A Multi-flow commodity and Constrained Shortest Path Model (MCSPM) presented in [198] is considered to find the optimal shortest routes for the multimedia flows from source to destination based on the defined specific constraints for a given service. The aim of MCSPM is to find a set of nodes with minimum link weight subject to multimedia flow constraints and perform better utilization of the network resources. As shown in [198], we also define the weight, We_{ij} of link (i, j) in this work as the sum of delay (dl_{ij}) and packet loss (pl_{ij}) values of a link multiplied with their scale factors $\alpha, \beta \geqslant 0$ respectively such that:

$$We_{ij} = \alpha \times dl_{ij} + \beta \times pl_{ij}, \forall (i,j) \in E \tag{6.1}$$

Table 6.1: Notation and Symbols Description.

Symbol	Descriptions
N	Total Number of DASH players
b_{ij}	Available bandwidth on link (i, j)
We_{ij}	Weight of link (i, j) calculated using equation 6.1
dl_{ij}	Delay on link (i, j)
pl_{ij}	Packet loss on link (i, j)
l_c	The link criticality
P^k_{max}	Max. acceptable packet loss of a subflow of an MPTCP connection k
D^k_{max}	Max. acceptable delay of a subflow of an MPTCP connection k
B^k_{sf}	The required bandwidth of a subflow sf of an MPTCP connection k
sf_k	Amount of subflow of an MPTCP connection k from $s \in V$ to $t \in V$
s_k	The source of a subflow of an MPTCP connection k
t_k	The destination of a subflow of an MPTCP connection k
$\alpha \geqslant 0$	Scale factor for delay
$\beta \geqslant 0$	Scale factor for packet loss
P_{st}	A set of the first h computed shortest paths from source to destination

While considering the MPTCP subflows of video service that have to be routed from MPTCP server to an MPTCP client, an optimization objective function is formulated to route all subflows in the network through shortest paths with minimized cost/weight as defined in equation 6.1. The objective function is defined as shown in equation 6.2:

$$O_f = min \sum_{(i,j)\in E} \sum_{k\in K} (We_{ij} \cdot sf^k_{ij}) \tag{6.2}$$

sf^k_{ij} indicates the subflow of a MPTCP connection k routed on link (i,j). Table 6.1 shows the parameters used to formulate the objective function constraints as used in this work.

The following constraints are formulated for the objective function shown in equation 6.2.
Constraint 1: To guarantee the subflow balancing in a softwarized network and ensure that, the total flow incoming into each vertex, V is equal to the total flow outgoing from the same vertex, with the exception of the MPTCP client and MPTCP Server, this work adopts the well-known flow conservation law as shown in equation 6.3.

$$\sum_{(i,j)\in E} sf^k_{ij} - \sum_{(j,i)\in E} sf^k_{ji} = \begin{cases} sf^k & \text{if } i{=}s_k \\ -sf^k & \text{if } i{=}t_k \\ 0 & \text{if } i \neq s_k, t_k \end{cases} \qquad \forall i \in E, \forall k \in K \tag{6.3}$$

Constraint 2: The maximum acceptable packet loss D^k_{max} imposes the limit for each subflow of an MPTCP connection k.

$$\sum_{(i,j)\in E} (pl_{ij} sf^k_{ij}) \leqslant P^k_{max}, \forall k \in K \tag{6.4}$$

Constraint 3: The maximum acceptable delay D^k_{max} imposes the limit for each subflow of an MPTCP connection k.

$$\sum_{(i,j)\in E} (dl_{ij} sf^k_{ij}) \leqslant D^k_{max}, \forall k \in K \tag{6.5}$$

Constraint 4: The available bandwidth for each link is bounded by an arc capacity constraint as defined in equation 6.6.

$$\sum_{(i,j)\in E} B^k_{sf} sf^k_{ij} \leqslant b_{ij}, \forall k \in K \tag{6.6}$$

A multimedia flow f is considered which consists of subflows $\{sf_1, sf_2, ..., sf_N\}$. In order to route these subflows of an MPTCP connection, the shortest paths between end-points in the network are calculated based on the link weight and bandwidth of the link as well as the QoE-demand of an MPTCP request. In real-world network operation scenario, a link may become a bottleneck when it can not provide the required resources or is likely to be the most used path for any pair of communicating nodes within an SDN/NFV network. The *link criticality* l_c parameter introduced earlier in section 5.6 is used to improve the routing performance based on the current load of each link e. The main objective of $c(l)$ as defined in equation 5.1 is to choose the links that can balance the loads across the network and avoid bottleneck links between pairs of communication nodes in softwarized networks. Using Fig 6.4 as an example, the link criticality for each link can be measured as follows. The first computed shortest path h=2 whereas the TD consists of three pairs of SDN/NFV switches (S1,S4), (S7,S4) and (S1, S8). The set of shortest paths P_{S1S4}, P_{S7S4}, P_{S1S8} are {S1→S7→S3→S4, S1→S7→S8→S4}, {S7→S3→S4, S7→S8 }, and {S4→S7→S8, S1→S7→S3→S8 } respectively. Since P_{S1S4} and P_{S1S8} pass twice through the link S1→S7, the link criticality of this link is 4/2=2. For link S7→S3, the link criticality is 3/2 because P_{S1S4}, P_{S7S4}, P_{S1S8} pass once through that link.

6.7.1 QoE-Driven Multipath Routing and Quality Optimization in Softwarized Networks

In order to formulate a QoE-driven quality optimization algorithm, an assumption is made that, during video streaming, a total of N segments each lasting for t_s seconds are available in one video sequence for download. Based on the available bandwidth, the MPTCP DASH client requests video segments that satisfy their quality levels. The set of quality levels requested are, Q_L = {Q_1, Q_2, . . . , Q_N}. The chain of bandwidths to download each video segment are, B_C = {b_1, b_, . . . , bN}. The MPTCP client can request an average video quality $V_{avg}(i,j)$ defined by equation 6.7.

$$VV_{avg}(i,j) = \frac{1}{N}\sum_{i=1}^{N} q_{i,j} \tag{6.7}$$

When the MPTCP client starts to download a segment i at t_s^i and the last bits of the segment is received at t_e^i, then the throughput during the downloading period of segment i is given by equation 6.8.

$$B(i) = \frac{S_{i,j}}{t_e^i - t_s^i} \tag{6.8}$$

where $S_{i,j}$ is the total bits in the segment $C_{i,j}$ with index i and bit-rate level j. The estimated available bandwidth that can be used to determine the bitrate level for the next chunk i to be downloaded is denoted by $\hat{B}(i)$. The estimated bandwidth can be calculated using an Exponentially Weighted Average (EWA) [199].

$$\hat{B}(i) = \hat{\alpha}B(i-1) + (1-\alpha)B(i-1) \tag{6.9}$$

According to [200], the value of α is fixed to 0.8. While buffer size Bf_{size} is very crucial for video rate adaptation in video streaming services, two buffer thresholds (Bf_{min}, Bf_{max}) are defined based in [199]. (a) if the MPTCP client is in buffering state, then Bf_{size} < Bf_{min}, (b) and Bf_{min}< Bf_{size} < Bf_{max}-S_d when the client is in playout state where S_d is the duration (2, 4, 6, 8 etc) of video segments in seconds. For the purpose of QoE evaluation using the proposed QoE-based multipath routing and quality optimization

algorithm, the initial delay, buffering events, stall duration and the bitrate level variation are considered. Before the initial delay I_{delay} of the video segment i is requested, then the initial delay of $I_{delay}(i-1)$ is accumulated during video streaming. Then if the bitrate level j is selected for i, the initial delay will increase by $\frac{S_{i,j,}}{\hat{B}(i)}$. From this formulation, the impact of the initial delay as defined in [199] can be given by equation 6.10.

$$I_{delay}(i,j) = (3.2)I_{delay}(i-1) + \frac{S_{i,j}}{\hat{B}(i)},\ Bf_{size} < Bf_{min} \tag{6.10}$$

To determine the impairments factors on video quality due to stalls, the combination of number of stalls and stall duration are considered. Note that, when the MPTCP client is in rebuffering state, the requested video segment i will introduces a stall duration by $\frac{S_{i,j,}}{\hat{B}(i)}$ whereas the buffer size is Bf_{size} + S_d - $\frac{S_{i,j,}}{\hat{B}(i)}$ when the segment i is downloaded. A threshold T_{ST} for stall indicators that avoid buffer length can be defined as:

$$T_{ST} = Bf'_{size} + \frac{S_{i,j}}{\hat{B}(i)} - S_d \tag{6.11}$$

Bf'_{size} is the warning threshold defined by γ Bf_{min} + (1-γ) Bf_{max} that shows the possible occurrence of buffering events. The number of stalls $N_{stall}(i,j)$ and stall durations $S_{drtn}(i,j)$ for the next video segment i with bitrate level j are defined by the following equations.

$$N_{stall}(i,j) = \begin{cases} N_{stall}(i-1)+1 & \text{if } Bf_{min} < Bf_{size} < T_{ST} \\ N_{stall}(i-1) & \text{Otherwise} \end{cases} \tag{6.12}$$

$$S_{drtn}(i,j) = \begin{cases} S_{drtn}(i-1) + \frac{S_{i,j}}{\hat{B}(i)} & \text{if } Bf_{size} < Bf_{min} \\ N_{stall}(i-1) + & \sum_{i+1}^{i+Bf_{min}} \frac{S_{i,0}}{\hat{B}(i)} \\ 0 & \text{if } i \neq s_k, t_k \end{cases} \tag{6.13}$$

The impairments indicators for stalls $I_{stall}(i,j)$ as defined in [201] can be calculated using a combination of $I_{delay}(i-1)$ and $S_{drtn}(i,j)$ as shown in equation 6.14.

$$I_{stall}(i,j) = (3.8)S_{drtn}(i,j) + 4.2N_{stall}(i,j) - 2.6\sqrt{N_{stall}(i,j)S_{drtn}(i,j)} \tag{6.14}$$

The bitrate level variation $B_{LV}(i,j)$ is another parameter which affect the end-user's QoE because of main two factors, namely, the video quality and the switching impact defined as $V_{Quality}(i,j)$ and $S_{impact}(i,j)$. The $V_{Quality}(i,j)$ is defined by equation 6.15. 3.8 and 4.2 are constants as defined in [201].

$$V_{Quality}(i,j) = \sum_{i=1}^{N} VQM_{l(i)} e^{\frac{0.02S_{drtn}S_d}{N}} \tag{6.15}$$

The $V_{Quality}(i,j)$ is the weighted mean of the $VQM_{l(i)}$ of the N chunks where $e^{(0.02S_{drtn}S_d)}$ is the weight of each video segments.

The $S_{impact}(i,j)$ is defined using the following equation.

$$S_{impact}(i,j) = \frac{1}{N}\sum_{i=1}^{N}[VQM_i - VQM_{i+1})^2)sign(VQM_{i+1} - VQM_i) \tag{6.16}$$

where

$$sign(x) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ otherwise} \end{cases} \tag{6.17}$$

The bitrate level variation $B_{LV}(i,j)$ is calculated as the sum of $V_{Quality}(i,j)$ and $S_{impact}(i,j)$ such that

$$B_{LV}(i,j) = B_1 \times V_{Quality}(i,j) + B_2 \times S_{impact}(i,j) \tag{6.18}$$

where B_1 and B_2 are the coefficients with values of 75.6 and 48.2 respectively [201]. The QoE metric $R_{QoE}(i,j)$ is calculated based on these three formulated factors, $B_{LV}(i,j)$, $I_{delay}(i,j)$ and $I_{stall}(i,j)$ using equation 6.19.

$$_{QoE}(i,j) = 100 - I_{delay}(i,j) - I_{stall}(i,j) - B_{LV}(i,j) + 0.17 I_{delay}(i,j)\sqrt{I_{stall}(i,j) + B_{LV}(i,j)} + 0.13\sqrt{I_{stall}(i,j) B_{LV}(i,j)} \tag{6.19}$$

To achieve a fair QoE allocation and an efficient resource utilization among MPTCP DASH clients, a QoE-driven multipath and quality optimization algorithm running over softwarized network is proposed that provides each MPTCP client with the video bit-rate level having the highest value of $R_{QoE}(i,j)$.

Algorithm 4: QoE-based Multipath and Quality Optimization Algorithm

input : flow f, *N_topology* G = (V,E)

1 *Compute link weight based on equation 6.1, bandwidth*
2 *Find all subflow shortest paths* $p \in P$ *in the network*
3 **foreach** $p \in Psrc \rightarrow dst$ **do** ;
4 **if** $p.used + sf_k.B^k_{sf} < b_{ij}$ **then return** p
5 **else** Go to step 3
6 *Perform mapping of subflow paths* p_{sf} *into SR paths*
output: List of Segment labels *SL*
7 Save path p with its associated list of *SL* in DB_p
8 *Start MPTCP subflow transmission based on client's QoE requirements*
9 Calculate buffer size Bf_{size} and $\hat{B}(i)$
10 Calculate $B_{LV}(i,j)$, $V_{Quality}(i,j)$ and $S_{impact}(i,j)$ using 6.18, 6.15 and 6.16
11 **if** $i < Bf_{min}/S_d$ **then**
12 Find $I_{delay}(i,j)$ using 6.10
13 **else** $I_{delay}(i,j) \leftarrow I_{delay}(i-1,j)$
14 **if** $Bf_{max} < Bf_{min}$ **then**
15 Find $N_{stall}(i,j)$, $S_{drtn}(i,j)$ using 6.12, 6.13
16 **else** Go to step 9
17 **end if**
18 **end if**
19 **end for**
20 **if** f_{new} *is a new flow* **then** Query DB_p to find the paths for subflows of f_{new}
21 **if** pf_{new} *is not in* DB_p **then** Go to step 2
22 *Issue path adding request for* f_{new} **end if**
23 **end if**
24 Continue transmission as long as $B^k_{sf} < b_{ij}$
25 Use the l_c and congestion index to avoid congestion

6.7.2 MPTCP Congestion Window Adaptation in Softwarized Networks

As described in section 5.2.3, the MPTCP client maintains a congestion window during video transmission to control the maximum number of packets to be sent at the destination point. As such, each TCP can trigger a decrease of its congestion window when it receives

duplicate ACKs because it interprets this as an indicator of a packet loss. The MPTCP client can reduce the traffic load of the transmitting path by halving its congestion window. However, coupled algorithms [174] such as LIA, OLIA and BALIA at the MPTCP flow level are used to control the TCP congestion window of all video subflows when it increases during video transmission. As shown in [202], the congestion window of each path may greatly differ from each other. This can lead to a large path delay difference which can affect greatly the throughput performance of MPTCP. It is important to mention that, when the difference of the delay between two paths increases, the time needed for MPTCP to receive all packets within the in-order unit also increases resulting into a decrease of the achievable network throughput. To improve the upper bound of goodput, then the E2E delay difference between transmission paths have to be minimized. In order to improve the system throughput of MPTCP, this work employs the congestion window adaptation algorithm presented in [202] that monitors the E2E delay of multiple paths.

To minimize the path delay difference, a delay ratio with lower and upper bounds is introduced. When the delay ratio increases, it triggers an MPTCP subflow to decrease its congestion window even if no packet loss is detected. This is not the case with the regular TCP which halves its congestion window when packet loss is detected in the network during transmission. Note that, the original MPTCP congestion window increase is considered to all video subflow. The delay ratio is defined as the ratio of the maximum path delay over the minimum path delay. The congetion window adaptation runs on the selected MPTCP shortest paths selected by the MPTCP module as shown in algorithm 4. Consider that a block of MPTCP N packets with Data Sequence Number (DSN) are sent from MPTCP client to the server. Suppose that, the destination point receives $N-1$ packets using path 1 and only 1 packet is received through path 2. This block of data is said to be transfered in an in-order unit. Suppose that P_t is the packet sending interval at the MPTCP client for transmission paths T_{paths} = {1, 2.....N}.

Fig 6.8 indicates an illustration of the MPTCP sequence Workflow when the DASH client that supports MPTCP requests a video file from the MPTCP server. Before video segments transmission starts from the server to the client, MPTCP connection is established using 7 steps as shown in Fig 6.8. For the purpose of demonstration, only two subflows are used. It is worth mentioning that, all subflows use the MPTCP congestion window adaptation algorithm presented in 4 which also runs on the proposed QoE-aware MPTCP/SR architecture in softwarized networks. The QoE management, SDN controller, MPTCP

Algorithm 5: MPTCP Congestion Window Adaptation Algorithm

```
1  if ϕ_min ϕ  ϕ_max
2  Select path of subflow i for CWND adaptation
3  i = arg max_p e2e of path p)
4  Adaptation counter does not exceed maximum limit
5  if Count_i< m then
6  Decrease congestion window of subflow i on path p
7  CWND_i ← CWND_i/ϕ
8  if Ssthresh_i>CWND_i then
9  Ssthresh_i = CWND_i
10 end if
11 Count_i ← Count_i + 1
12 else  Reset adaptation counter
13 Count_i = 0
14  then
15 Return an improved network throughput
```

module and SR module perform their respective functions as described in section 6.6. The difference with the approach presented in chapter 6 compared to that of chapter 5 is that, the congestion window of MPTCP subflows is modified to achieve more network throughput. However, the coupled OLIA congestion window is applied for all subflows as presented in chapter 5.

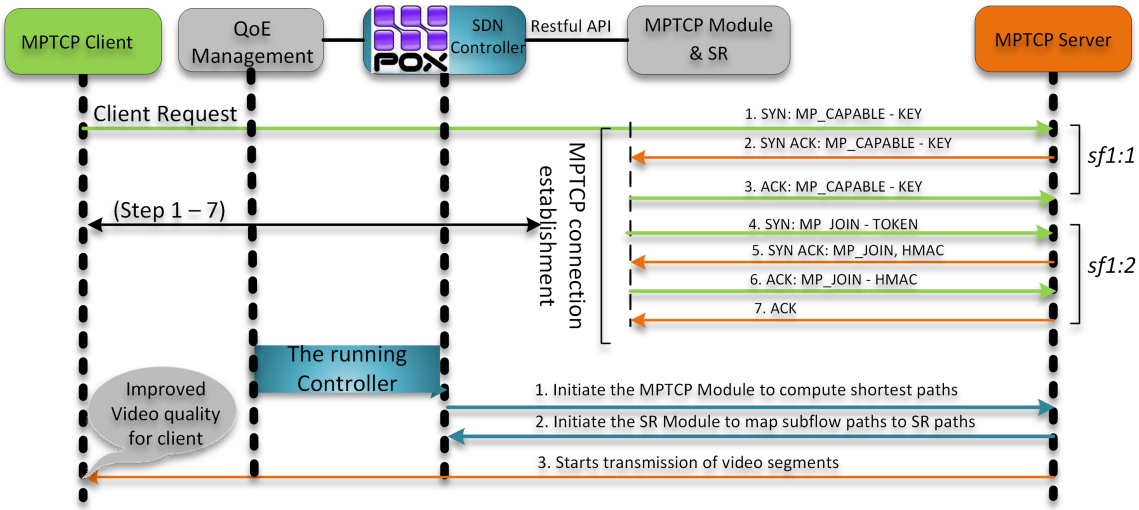

Figure 6.8: MPTCP sequence workflow diagram

For path protection and recovery mechanisms, an assumption is made that when link $(i,j) \in E$ fails during a normal network operation, then the controller has to recompute new shortest paths after pruning the failed link (i,j) (see Fig 6.5). This procedure is done after performing a pair of robustly disjoint SR-paths, using a depth first search. The path protection and recovery algorithm requires information on the SR-path latencies in order to prune the search. The SR-path latency is represented as a L: $n \times n$ matrix such that

$L(x,y)$ indicates the minimum latency path from x to y in G. This matrix is recomputed every time the link failure occurs to find the SR-path with minimum packet loss, E2E latency and delay on a link. Suppose $p_f(s, t)$ indicate the shortest path consisting of the failed link. This work denotes h_{New} as the set of new shortest paths. The controller is required to perform normal procedures for mapping these new paths to SR paths for transportation of MPTCP subflows from source to destination. Algorithm 6 is used to compute new shortest paths during link failure.

Algorithm 6: Path Protection and Recovery of Link Failure (PathReLieF)

1 *Remove failed link*(i,j)
  input : New_topology $G' = (V',E')$
2 *Find all subflow shortest paths* $p' \in P'$ *in the network*
3 **foreach** $p' \in P'$ **do if** p'*.used* $+sf_k.B^k_{sf} < b_{ij}$ **then return** p'
4 **else** Go to step 3
5 **end if**
6 Map all subflow shortest paths p'_{sf} into SR' paths **output:** List of Segment labels SL'
7 Go to step 6 in algorithm 3
8 **end for**

6.7.3 Experimental Testbed & Setup

In order to evaluate the performance of the proposed architeture and the developed algorithms, an experimental testbed shown in Fig. 6.9 is used. The tesbed consists of the edge, aggregation and core layer which consists of 8, 4 and 2 SDN switches respectively. Two VMs both running Linux (Ubuntu V16.04 LTS) were installed with the MPTCP v0.92. The mininet that used to model the network redundant links at each level as shown in Fig 6.9 was installed in another VM. Redundant links are used in datacenter networks to enable the implementations of multipath routing in SDN/NFV systems. The main objective of using a topology with redundant links as shown in the figure is to increase network availability, reliability and avoid network failures that can negatively affect the performance of the network and the end-user's QoE. As such, when one link/node fails in a softwarized network, then another links/node could establish a connection and transfer the video flows from MPTCP client to the server and vice versa. The POX controller was installed in the second VM. The SR module was implemented following a customization of source codes available at [203].

Table 6.2: A list of used parameters

N.Layer	Bandwidth (Mbps)	Delay (ms)	Packet Loss (%)
Edge	3	15	1
Aggregation	2	20	3
Core	3	30	3

6.7.4 Video Clips/Sequences Used for Experiments

The "Big Buck Bunny" video clip shown in Fig. 5.10 was encoded using *ffmpeg* version 3.3.4 with the *libx265* at 3 different resolutions (720p, 480p and 360p) with video encoding rate of 2.496Mbps, 1.536Mbps and 1.0888Mbps respectively. The link parameters configured in Fig. 6.9 are shown in Table 6.2. When the MPTCP is enabled, the default configurations of MPTCP V0.92 is used and the MPTCP path manager is configured to a *full-mesh* to limit each MPTCP connection to have only 3 MPTCP subflows. The VLC-DASH plugin [180] is used as the DASH client for collecting and reporting the performance of video quality and throughput measurements during video streaming. The system throughput is computed from Payload_bits/download_time where a *Payload_bits* is the number of extracted bits of the video content per single unit time. Apache Server was installed on two machines that supports MPTCP. These machines were then attached on mininet network as shown in Fig 6.9. The video transmission is repeated 40 times from MPTCP server to MPTCP client. The proposed QoE-aware MPTCP/SR-based architecture for softwarized networks is compared with the MPTCP and regular TCP in terms of throughput and the end-users-QoE.

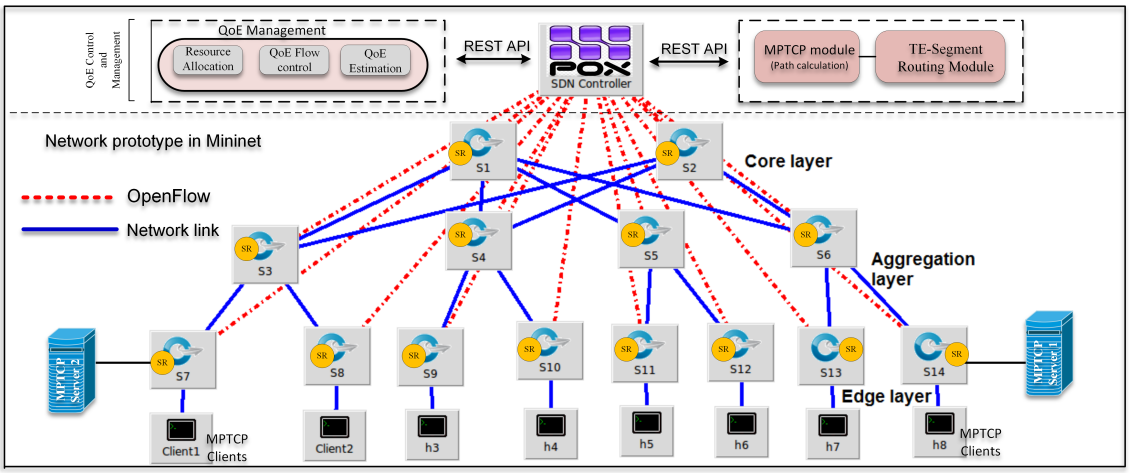

Figure 6.9: QoE-aware MPTCP/SR experimental testbed and prototype used

6.8 Experimental Results and Discussion

6.8.1 System Throughput

Fig 6.10 shows the comparison of system throughput of the QoE-aware MPTCP SDN/NFV SR-based proposal, MPTCP and the regular TCP for video resolution of 360p. Note that, chapter presented the comparison results of QoE-aware MPTCP/SR SDN/NFV based approach with TCP. This work includes the investigation of MPTCP performance with both TCP and the proposed approach under the same experimental settings. Note that, this is a shortcoming of chapter 5 which only compared the performance of TCP and QoE-aware MPTCP SDN/NFV SR-based strategy. For each video transmission, the throughput of the proposed approach is higher compared to MPTCP and the regular TCP. The performance of the proposed QoE-aware MPTCP/SR approach is better it exploits multipath transmission and employ SR which does not require any path signaling. This is not the case with MPTCP and the regular TCP which does not use SR mechanisms during transmission of video flows from the MPTCP client to the server. For example, at the 3min, the throughput achieved for the QoE-aware MPTCP/SR, MPTCP and the regular TCP is 1.82Mbps, 1.71Mbps and 1.62Mbps. Apart from using multiple paths during transmission in the proposed approach, the subflows also use the congestion window adaptation algorithm provided in 15.

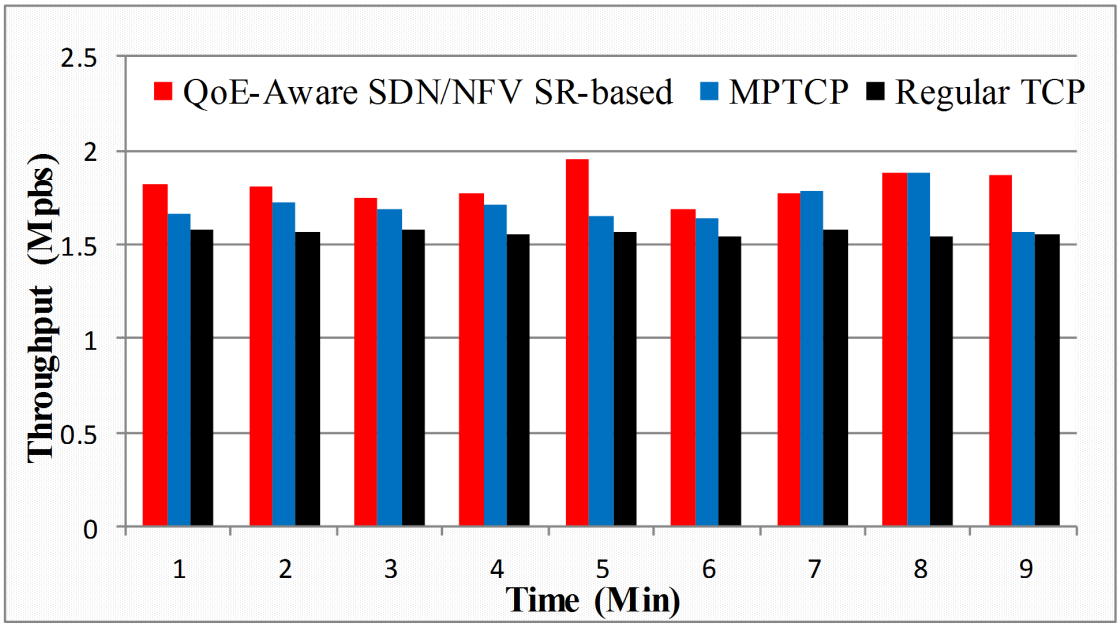

Figure 6.10: Comparison of system throughput for a video resolution of 360p

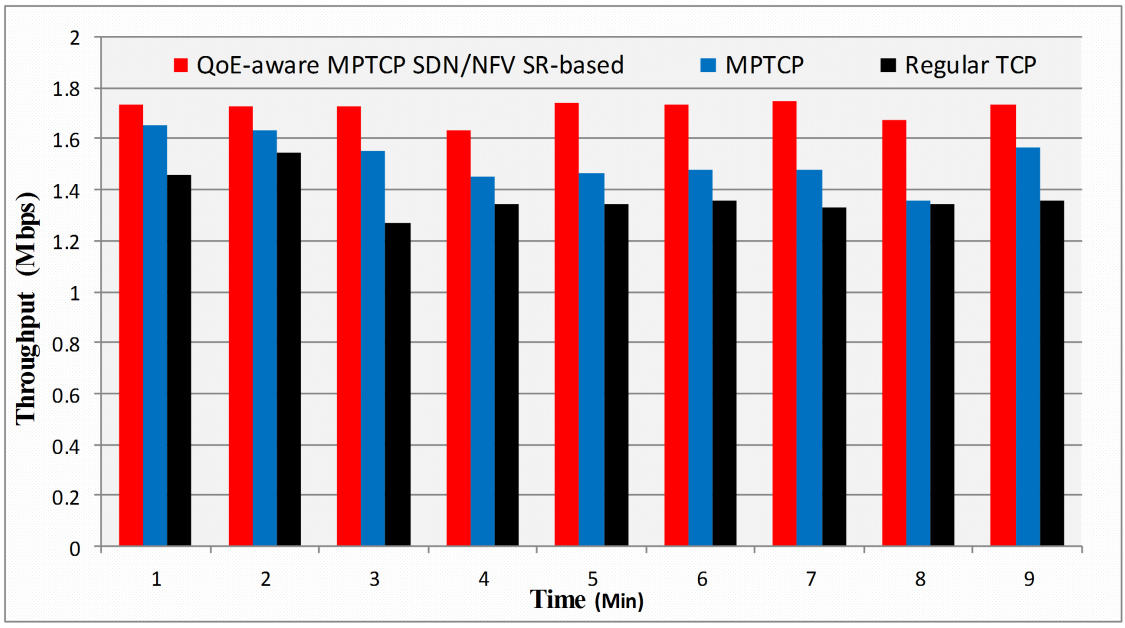

Figure 6.11: Comparison of system throughput for a video resolution of 480p

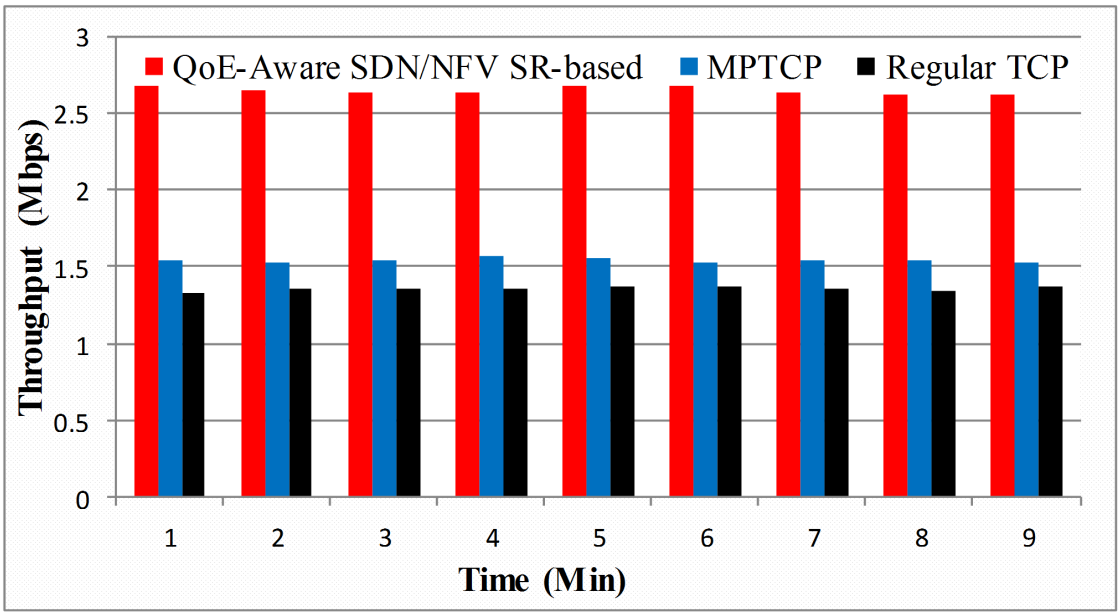

Figure 6.12: Comparison of system throughput for a video resolution of 720p

The QoE-aware MPTCP/SR and MPTCP performance becomes similar at some points during video streaming because of using multiple paths during transmission. As shown in Fig. 6.10, this happens at the 7min and 8min where the achieved system throughput is 1.80Mbps for both QoE-aware MPTCP/SR and MPTCP approaches. However, at the 9min, the MPTCP and regular TCP performance is similar because of Table 6.3 shows the average throughput achieved by QoE-aware SDN/NFV SR-based, MPTCP and TCP for 360p, 480p and 720p after 40 runs of video streaming.

Table 6.3: Average throughput in Mbps

Video Resolution	QoE-aware MPTCP/SR	MPTCP	TCP
360p	1.87	1.73	1.55
480p	1.76	1.61	1.36
720p	2.66	1.56	1.37

6.8.2 Video Reception Quality Measurements

For video quality, the reception quality (ρ) [204] metric defined as the ratio between download throughput and video encoding rate is used for evaluating the performance of the proposed algorithm. If (ρ) >1 then the video has good reception quality otherwise, the video has poor quality. Fig 6.13, 6.14 and 6.15 shows the comparison of a video streaming reception quality using resolution of 360p, 480p and 720p respectively. It is clear that, the QoE-aware MPTCP/SR SDN/NFV-based approach performs better compared to MPTCP and regular TCP for all three used video resolutions.

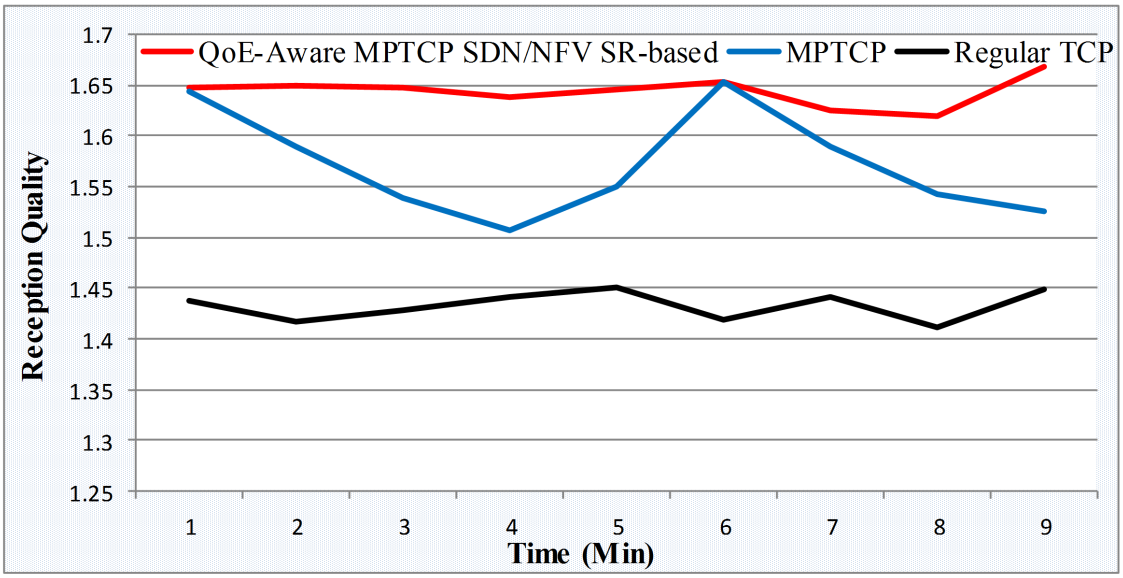

Figure 6.13: Comparison of video streaming reception quality for 360p

6.8.3 Measurements of Failure Recovery and Localization Time

In order to implement the path protection and link recovery algorithm proposed in Algorithm 6, one link from the Mininet topology shown in Fig. 6.9 was removed using the pseudocodes shown in Algorithm 7. The link connecting S1 and S4 was purposely

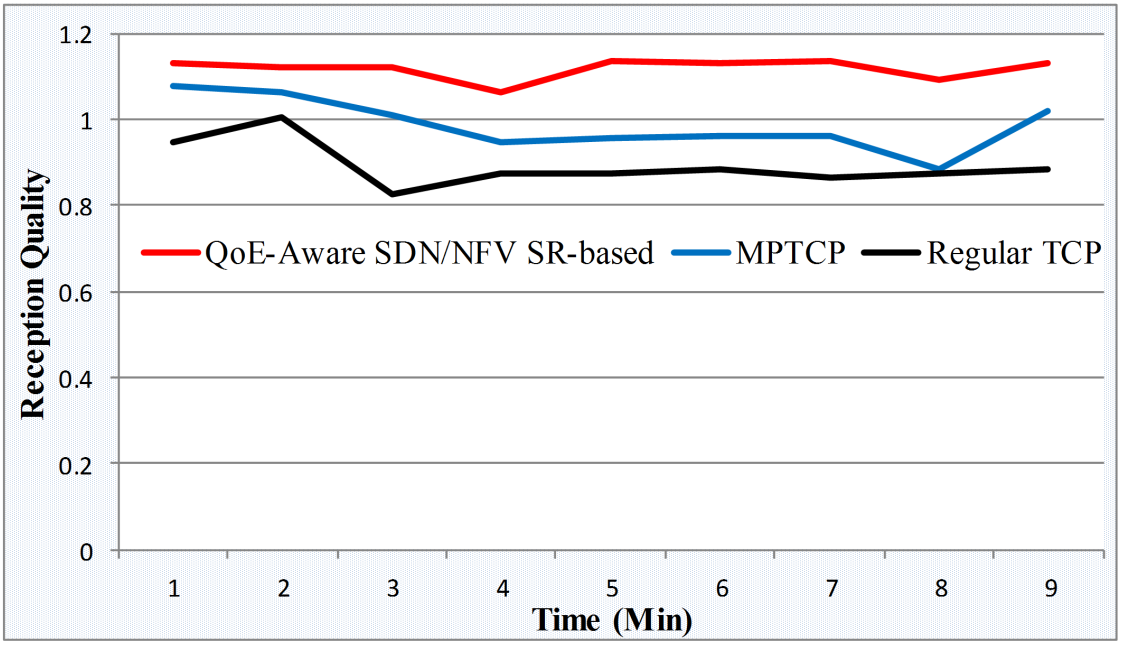

Figure 6.14: Comparison of video streaming reception quality for 480p

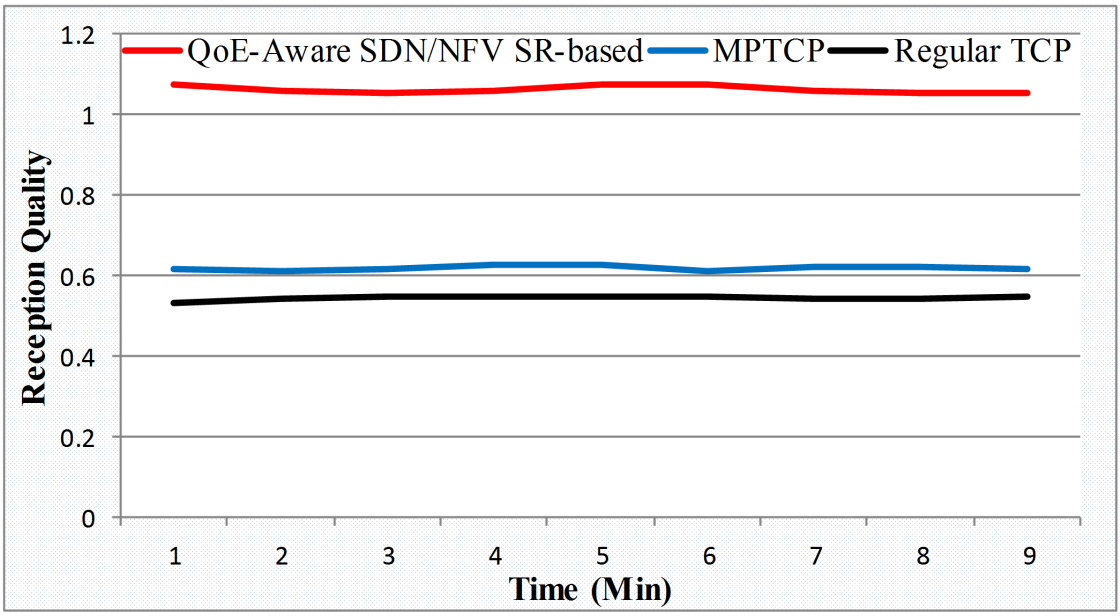

Figure 6.15: Comparison of video streaming reception quality for 720p

disabled so that new multiple shortest paths can be computed by Algorithm 6. In SDN, the controller is solely responsible to detect network failure and through Operations Administration, and Maintenance (OAM) tools. Link or node failure detection in SDN becomes more efficient because of the global view of the SDN controller which enables OpenFlow switches not to flood the network with messages for topology synchronization. The OAM tools provide performance indications that can help to identify the problem area in the network and isolate the root cause of a problem. The existing OAM tools in SDN

Algorithm 7: Pseudocode to Remove a Link from the Network Topology

```
1 def removeLink(self,node1, node2, port1=None, port2=None, **opts):
2 if not opts and self.lopts: then
3 self.addPort(node1, node2, port1, port2)
4 key = tuple(self.sorted([node1, node2]))
5  self.link_info[key] = opts
6  self.g.remove_edge(*key)
7 return key
```

can provide tests for continuity check, connectivity verification, link trace, delay and loss ratio. The tools exist in layer 2 and layer 3 and SDN can use the OAM tools like Link Layer Discovery Protocol (LLDP). Currently, the SDN controller can compute disjoint paths using path calculation engines. For example, Opendaylight's topology processing includes Suurballe algorithm [205] implementation to calculate disjoint paths in the network. The main idea of Suurballe algorithm is to use Dijkstra's algorithm to find one path, modify the weights of the graph edges, and then to run Dijkstra's algorithm a second time. In POX controller, the failure detection is conducted using the POX Topology Discovery module [206]. When a link associated with the Openflow switches is detected as having been removed or failed, a *LinkEvent* message is raised by the Topology Discovery module. The *openflow.discovery* component in POX controller sends LLDP messages out of OpenFlow switches so that it can discover the network topology. However, the controller considers that the link is timed out or failed when the switch does not send the LLDP packet associated with the port connected to its neighbor. In this circumstance the controller raises the *LinkEvent* for link removal.

Fig. 6.16 indicates an example of path protection and link recovery mechanisms in softwarized networks based on the proposed algorithm 6. On failure detection in SDN, crankback routing [207] is normally applied where the traffic is recursively returned back to the source node from which the traffic can be forwarded to the destination node again. However, the crankback routing increases network recovery time because the failure detection windows also increase due to the longer path lengths. The proposed mechanism using algorithm 6 does not need crankback routing and the failure detection window, thus the recovery time is minimized. As shown in Fig. 6.16, upon link or node failure, the data packets of video flows transmitted are sent back to the original path. The node where the link failure is detected reroutes the data packet until the convenient route is found where the packets can be forwarded to their destination point. The novelty of this

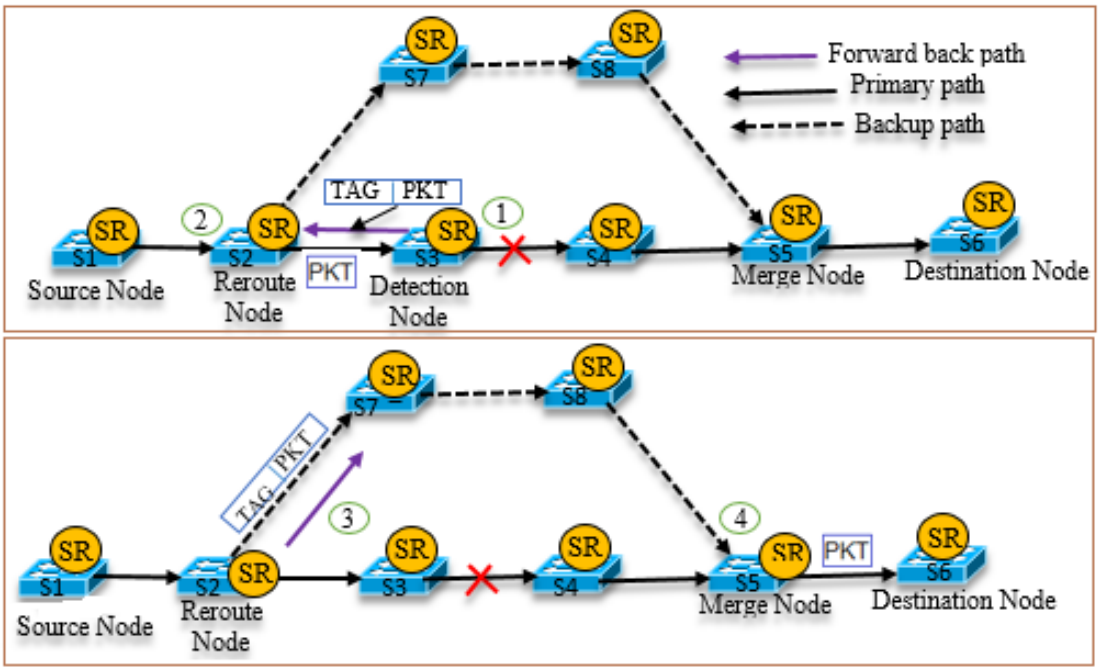

Figure 6.16: An example of path protection and link recovery mechanisms in softwarized networks

approach is that, the same data packets of video flows are tagged first (e.g. with a SR label which contains information on the failed link) and then sent back through the primary path. After receiving the tagged packet, a reroute node (e.g., S2) can then respond to the link/node failure and forward the tagged packets to their destination node. When the reroute node processes the first tagged packet, a state transition is performed in the OpenFlow switch. A failover table is created as shown in Fig. 6.5 and all subsequent packets coming from the source node are forwarded on the reroute node. The proposed path protection and link recovery algorithm is designed to reduce path costs, network recovery time and backup path length.

In order to evaluate the performance of the proposed algorithm, the links between each pair of switches were failed by using the pseudocode shown in algorithm 6. The failed link between each pair of switches were repeated 20 times per pair. The failure recovery time, backup path length and failure localization time were collected to investigate the performance of the proposed algorithm when the failure is detected. Failure localization is the process of identifying the exact point where failure has happened. The Failure Localization Time (FLT) is the time taken by the SDN controller to get the link failure information from the POX and OpenDaylight *Topology Discovery* module to the time it finds the exact failure location in the network. The Failure Recovery Time (FRT) is the time taken

by the SDN controller to receive the information regarding the exact point of link failure and install the new flow failover tables in Openflow switches. The proposed algorithm 6 for path protection and link recovery was implemented in POX and OpenDaylight controllers. The main idea was to investigate on how the two SDN controllers react to link/node failures in the network in terms of failure recovery and failure localization time.

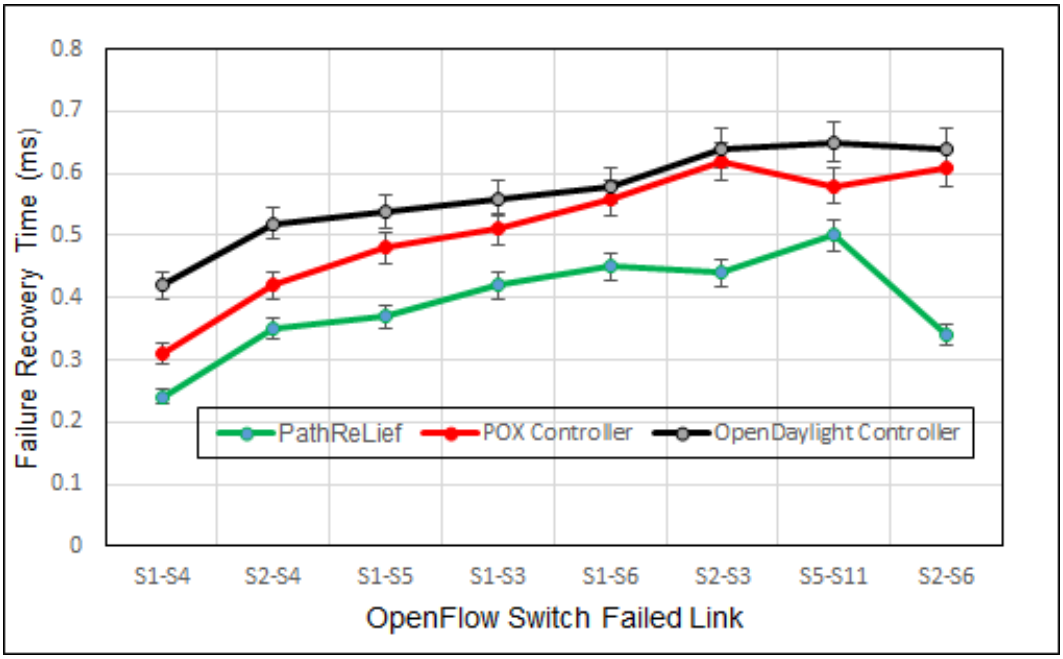

Figure 6.17: Performance comparison in terms of failure recovery time

Failure Recovery Time (FRT)

Figs. 6.17 and 6.18 show the performance comparison with and without PathReLief algorithm using the POX and OpenDaylight controllers. The performance of PathReLief is better than that of native algorithms implemented in POX and OpenDaylight. This is so because, not only PathReLief uses MPTCP for transmission but it also applies SR technology on every OpenFlow switch which makes the transfer of data from the failure node to the SDN controller fast. It is important to mention that, the SR is disabled while making while investigation the failure recovery and localization time using POX and OpenDaylight controllers. It can be observed from Fig. 6.17, when {S1→S4} fails, the PathReLief takes to only 0.24ms to perform a failure recovery in the network and install failover tables. Note from the network topology shown in Fig. 6.9, link {S1→S4} connects switches belonging to the core layer and aggregation layer which are closely connected to the SDN controller. This also contributes to the minimization of time required to perform

link recovery in the network. The failure recovery time when {S5→S11} fails is the largest (0.5ms, 0.58ms, 0.65ms) for PathReLief, POX and OpenDaylight respectively. This is because link {S5→S11} connects switches belonging to the aggregation layer and the edge layer. This makes the backup path length from the controller to failed node to be large making the recovery time to increase. Fig. 6.17 also indicates that, the POX controller performs better compared to OpenDaylight controller in terms of recovery times for all failed links in the network. This happens because the POX controller have shown to have a minimum response time and a reduced installation time of flow rules as demonstrated by [208]. It is worth mentioning that, the proposed algorithm performs better than the native implementations of POX and OPneDaylight controller.

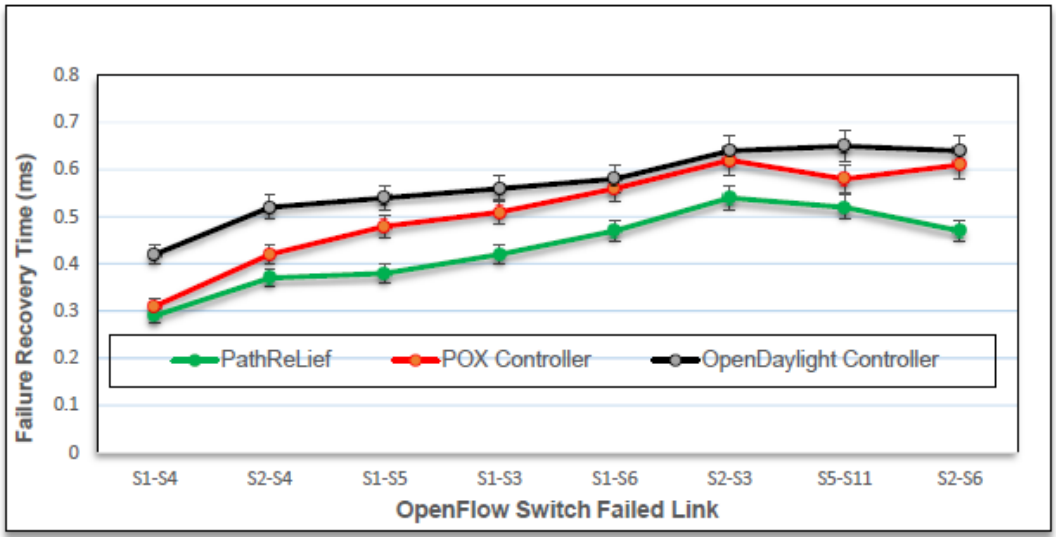

Figure 6.18: Performance comparison in terms of failure recovery time using OpenDaylight controller

Failure Localization Time (FLT)

Fig. 6.20 indicates the performance of the failure localization time for PathReLief, POX and OpenDaylight controller using the same failed links from the network topology shown in Fig. 6.9. It can be observed that, the PathReLief performs better than the POX and OpenDaylight controller. For example, when {S1→S4} fails, the failure localization time of PathReLief, native discovery module in POX and OpenDaylight are 48µs, 81µs and 94µs respectively. The time taken for the controller to detect the exact point of a failed link is shorter for PathReLief because {S1→S4} connects the core and the aggregation layer in the fat-tree topology shown in Fig. 6.9. For failed link {S5→S11}, the failure localization is 78µs, 85µs and 96µs. The localization is large because link {S5→S11} connects the aggregation

layer and the edge layer and therefore making the backup path length to increase.

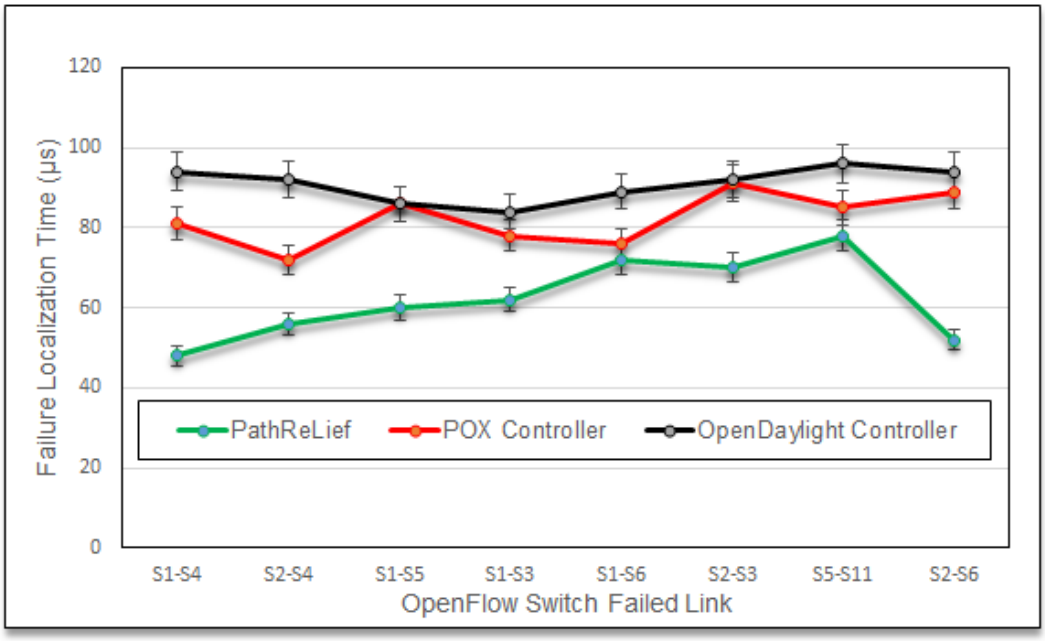

Figure 6.19: Performance comparison in terms of failure localization time

Link Capacity Occupation (LCO)

The LCO is calculated as the percentage of the total link capacity allocated to the backup paths considered for forwarding the tagged packets. When a failure of a link or a node occurs in the network, some of the available bandwidth are allocated to the backup path depending on the network traffic that are to be sent from the failed node/link to the destination node. The PathReLief shows a minimum link capacity occupation allocated

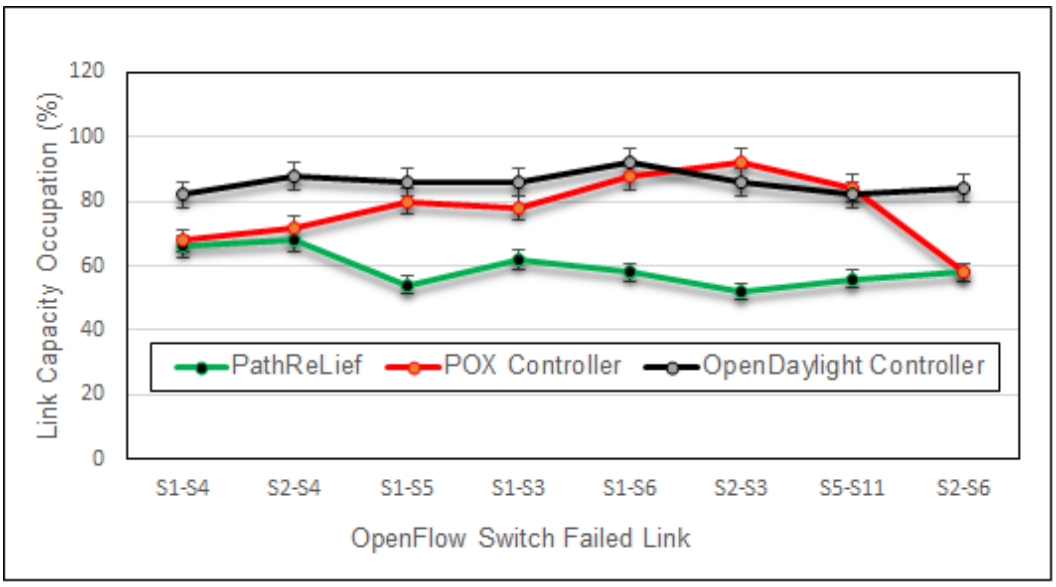

Figure 6.20: Performance comparison in terms of link capacity occupation

to the backup paths compared to POX and OpenDaylight controller. This is so because MPTCP and SR would enable load balancing between multiple disjointed paths and therefore making the transmission of the tagged packets with minimum bandwidth fast on every link used in the network. The PathReLief would also allow the MPTCP clients to share the bandwidth in the network and remove self behaviour encountered in tradition networks.

Table 6.4 summarizes the performance of PathReLief, POX and OpenDaylight in terms of failure recovery time, failure localization time, the primary path ratio and the Link Capacity Occupation (LCO). The LCO is calculated as the percentage of the total link capacity allocated to the backup paths that was considered for forwarding the tagged packet. The Primary Path Ratio (PPR) is used to measure the capacity of network recovery to reset the connection between pairs of disconnected nodes. When failure occurs in the network, some of the available bandwidth are allocated to the backup path depending on the video flows that are to be sent from the failed node/link to its destination node. The PathReLief shows a minimum number of the capacity allocated to the backup paths compared to POX and OpenDaylight controller. This is so because MPTCP and SR would enable load balancing between multiple disjointed paths and teherefore making the transmission of the tagged packets with the minimum bandwidth fast on every link used in the network.

6.9 Summary

In this chapter, a QoE-aware MPTCP/SR-enabled architecture that can achieve an optimized E2E QoE-level for the end-users is presented. To improve the video quality, the author propose to use multiple shortest paths for MPTCP subflows transmission and choose important intermediate nodes to perform source routing using SR paradigm. The aim is to meet future networks bandwidth aggregation and provide an efficient orchestration, QoE control and management of future multimedia services in future networks (e.g., 5G). Preliminary results shows that, the proposed approach outperforms the MPTCP and regular TCP in terms of system throughput and the end-user's QoE. Moreover, this chapter propose a path protection and dynamic link-recovery approach using MPTCP and SR to increase survivability, resilience, availability of services in future 5G networks. To demonstrate the effectiveness of the proposal, the performance of the

Table 6.4: A Summary of Performance Comparison for Path Protection and Link Recovery Mechanisms in Softwarized Networks

Method	Link	FRT (ms)	FLT (μs)	Link Capacity Occupation (%)	PPR
PathReLief	S1→S4	0.24	48	66	1.00
	S2→S4	0.35	56	68	1.00
	S1→S5	0.37	60	54	1.00
	S1→S3	42	62	62	1.00
	S1→S6	0.45	72	58	1.00
	S2→S3	0.44	70	52	1.00
	S5→S11	0.5	78	56	1.00
	S2→S6	0.34	52	58	1.00
POX	S1→S4	0.31	81	68	1.00
	S2→S4	0.42	72	72	1.00
	S1→S5	0.48	86	80	1.00
	S1→S3	0.51	78	78	1.00
	S1→S6	0.56	76	88	1.00
	S2→S3	0.62	91	92	1.00
	S5→S11	0.58	85	84	1.00
	S2→S6	0.61	89	58	1.00
ODL	S1→S4	0.31	81	82	1.00
	S2→S4	0.42	72	88	1.00
	S1→S5	0.48	86	86	1.00
	S1→S3	0.51	78	86	1.00
	S1→S6	0.56	76	92	1.00
	S2→S3	0.62	91	86	1.00
	S5→S11	0.58	85	82	1.00
	S2→S6	0.61	89	84	1.00

proposed PathReLief algorithm and the conventional topology discovery mechanisms are compared. Preliminary results show that, PathReLief outperforms others used in the commonly used controllers (i.e., POX and OpenDaylight), in terms of reduced failure recovery time and localization time.

Chapter 7

Discussions, Future Work and Conclusions

7.1 Introduction

Future networks such as 5G are expected to provide and support a wide range of mobile applications and multi-variety of services with QoE guarantees. While video -based services are among the most critical services supported in current 4G LTE and future networks, a lot of efforts need to be done so as to improve and optimize the video quality using mechanisms such as network resource allocation, routing, server selection and video compression optimization. As such, the end-user' demand for services with better quality from service providers has triggered the trend of traditional and future networks towards using QoE in network management through an efficient utilization of network resources. In order to meet end-users' QoE requirements and their expectations, several QoE issues (e.g., QoE monitoring, QoE control and management) pose challenges to service providers. Such issues are contributed by existing technologies which are either unable to adapt to diverse changing network conditions, or limited in the available resources. In addition, existing QoE adaptation approaches limit the network ability to provide intelligent and efficient solutions such that they only react when a problem occurs and therefore resulting in sub-optimal network performance. These challenges necessitate harmonizing the integration of advanced intelligence architectures (e.g., using SDN and NFV) with the development of intelligent QoE-aware adaptation mechanisms that will optimize and improve the end-users' QoE. The main objectives of this work are to (2) develop a video quality management approaches based on the traffic intensity under DASH using SDN, (3) investigate and develop a QoE-driven SDN based resource allocation mechanisms for

improving the end-users' video quality, (4) propose and develop a novel QoE-aware, SDN-based approach for managing video streaming services using MPTCP and SR paradigms, and (5) develop new algorithms and architectures for QoE control and management of video streaming services in softwarized networks that are enabled by MPTCP and SR. It is worth mentioning that, MPTCP and SR are utilized as promising application protocols and data transfer technologies to achieve an optimized E2E QoE level for the end-users in future softwarized 5G network. The extensive simulations performed in this work, quality evaluations based on data collected from from real-time multimedia transmissions over the developed SDN/NFV platform demonstrates that the proposed algorithms can reduce the number of stalls and their duration by more than 84% and 94% compared to the baseline native DASH.js. This Chapter provides an extensive discussions of the main contributions of this work and highlights the novelty, future research directions in the field of QoE control and management in softwarized networks and the conclusions.

7.2 Contribution to Knowledge

The main contributions presented in this thesis are discussed in the following subsections.

7.2.1 QoE-based Multimedia Flow Routing Mechanisms using SDN/NFV

This chapter makes two contributions regarding the QoE management using flow routing mechanisms in SDN/NFV-based networks, this thesis makes two contributions. First, a video quality management scheme over an SDN architecture is proposed based on traffic intensity. The proposed algorithm can mitigate virtual port queue congestion which may cause buffering events and reduce the video quality. The main goal of the proposed scheme is redirecting traffic flows to less congested ports in the network. Three video clips categorized as fast movement, medium and slow movement were used for experiments. The YouTube API player is used for collecting relevant information such as buffering duration, the number of stalls and duration of stalls during video streaming. The number of stalls and stall duration for DASH with the proposed scheme are measured and compared with the performance of the native DASH.js. The experimental results indicate that, the proposed video quality management scheme demonstrates effectiveness of mitigating the drawbacks of DASH by significantly reducing the number of stalls and their duration by more than 84% and 94%. Note that, our work is based on transmitting

video flows using one path from source to destination. The work has been contributed to the research community following the publication achieved in [26]. The contributions in this work is described in Chapter 3.

7.2.2 QoE-driven SDN based Resource Allocation Algorithm in Future Networks

Enabling the dynamic configuration and resource allocation in SDN-based networks can be significant and an outstanding approach on how resources are used for the purpose of improving the end-users' QoE. This would also enable coping with the continuous growth of multimedia applications in future networks such as 5G. In this aspect, a novel QoE-driven resource allocation mechanism that assigns tasks dynamically to virtual network nodes in order to achieve an optimized end-to-end quality is proposed in this chapter. Network softwarization and virtualization technologies (e.g., SDN and NFV) were applied to provide an efficient control and management of resources. The main goal of the proposed strategy is to find the best combination of network node functions that can provide an optimized QoE level to end-users through node cooperation. The video streaming service is taken into account as a collection of tasks where a good combination of switches (S1, S2...) is determined and negotiate the assignment of these tasks by considering the end-users' perceived final video quality.

This work employs the task-level scheduling with the intention of assigning tasks related to multimedia services to network elements. To achieve this objective, a QoE-driven dynamic task allocation algorithm for adaptive video streaming was designed and implemented over SDN/NFV enabled networks. The proposed approach indicates that, the capabilities provided by SDN and NFV are significant for enhancing the video quality, allocate and manage network resources efficiently in future softwarized networks. Based on the proposed QoE-driven resource allocation, service providers can meet future changing business goals while succeeding high level of end users' QoE. Experimental results indicate that, the proposed strategy can improve the video quality when the normalized QoS values decreases (low packet loss and delay). The work has been contributed to the research community following the publication achieved in [27]. The contributions in this work is described in Chapter 4.

7.2.3 A Novel QoE-Centric Multipath Routing Algorithm for Video Streaming using SDN

This work extensively explores the use of multiflow strategy where multiple disjointed paths are used in SDN-based networks to improve network resources utilization and end-user's QoE for delivering video streaming services. This chapter provides the first implementation mechanisms of MPTCP and SR for traffic management in software defined and virtualized networks. The contribution under this work propose a novel QoE-aware SDN-based MPTCP/SR framework to enhance the QoE for video streaming services delivery in future networks. In the proposed framework, the POX controller was extended by introducing three modules namely, (a) MPTCP-flow manager, (b) QoE management, and (c) segment routing (SR) module. A QoE-centric multipath routing algorithm that forwards multimedia flows through multiple disjointed paths using SDN controller is also proposed. The idea behind the proposed algorithm is to perform source routing to video subflows using SR paradigms. DASH is employed to test the performance and effectiveness of the proposed algorithm. This scheme is compared with the traditional TCP approach. The QoE-aware multiflow approach based on SDN significantly improve network throughput and link utilization by 50%. It also reduces significantly the number of video quality and startup delays. The work has been contributed to the research community following the publication achieved in [63], [29]. The contributions in this work is described in Chapter 5.

7.2.4 QoE Control and Management of Video Streaming Services in Softwarized Networks

A centralized SDN controller is a fundamental element for routing the MPTCP subflows from one point to another through multiple paths. This work provides the first realization of TE management strategies where MPTCP and SR are further employed in softwarized networks to facilitate efficient transfer of large amount of multimedia applications between end-points. It extend the contribution four (4) described above, where MPTCP and SR were utilized to improve video quality and the system performance for video streaming services. In order to improve the video quality, the Multi-flow commodity and constrained Shortest Path Model (MSPM) was used. The aim of MSPM is to select intermediate nodes and perform source routing using SR paradigm. A novel QoE-aware

MPTCP/SR-enabled architecture in softwarized networks is proposed to provide an efficient orchestration, QoE control and management of multimedia services delivery over future networks. Moreover, a system model and a QoE-based multipath and video quality optimization algorithm called "QoEMuSoRo" are proposed to forward traffic using SR paradigms over the proposed SDN-enabled NFV-based system. Note that, the proposed architecture not only provide multiple disjointed shortest paths for MPTCP subflows and perform source routing using SR paradigm but it can also adapt to changing network conditions or media contents. This is an important aspect for service providers and mobile operators for improving the end-users' QoE through efficient control and management of future multimedia services. The preliminary results show that, the proposed approach outperforms the MPTCP and regular TCP in terms of network throughput and the end-user's QoE. The work has been contributed to the research community following the publication achieved in [30]. The contributions in this work are described in Chapter 6.

7.2.5 Multipath Protections and Dynamic Link Recovery in Softwarized 5G Networks using Segment Routing

The performance of future softwarized 5G infrastructures can be affected by link or node failures which are inevitable occurrences in the networks. In SDN/NFV-based network, not only the centralized controller itself can be a single point of failure but also the data plane. If the SDN controller or link/node fails, then the routing and forwarding capabilities of the 5G network become down and can lead to undelivered data packets, unreliable network and even drop new flow requests from end-users. To overcome this, future 5G softwarized networks should be robust enough to ensure high reliability and availability of services by making sure that any failed parts in the network are detected, restored and recovered within a permissible period of time and at the lowest achievable cost. Future softwarized 5G networks have to be robust enough so as to ensure high network reliability and services availability. 5G network architecture have to ensure that any failed parts in the network are detected, restored and recovered within a permissible period of time and at the lowest achievable cost. The contributions of this part are two fold: (a) a multipath protections and Link- Failure free MPTCP/SR-based architecture that increases survavibility, resilience, availability and robustness of 5G networks is proposed, and (b) a system model and a multipath protections and dynamic link- failure free algorithm called "PathReLief" is introduced to reduce failure recovery time and avoids link congestion in

MPTCP/SR SDN/NFV 5G networks. Extensive experiments are conducted to evaluate the performance of the proposed PathReLief algorithm over the developed softwarized network platform. The results indicate that PathReLief outperforms the conventional topology discovery mechanisms used in the POX and OpenDaylight controllers in terms of reduced failure recovery time and localization time.

7.3 Limitations of this Work and discussions

This work has a number of limitations which can be investigated in future work as described below.

7.3.1 The use of Open Source Tools for Simulations

The main advantage of open source tools is that, they are customizable, free to use and save costs. However, one of the drawbacks of using open source tools is compatibility because their versions keep changing because of supporting customization. For example, the algorithm developed using Mininet V.2.2.O and OpenDaylight (Oxygen-SR4) needs to be customized so that it can run on previous versions of Mininet and OpenDaylight. The performance of the proposed algorithms such as video quality management presented in Chapter 3, QoE-driven resource allocation and multipath routing using MPTCP and SR presented in Chapters 3, 4, 5 and 6 respectively are assessed using open source tools such as Mininet and Opendaylight SDN controller. Although this approach provides a lots of benefits such as easy configurations, repeatable and fast but some of the configured parameters such as jitter, packet loss and the network bandwidth were made to be controllable throughout the experimentation period. However, moving to physical implementations for example in data center networks where these parameters are unpredictable, the proposed algorithms could provide the same performance but with some complexity because of the variability of these network parameters in real world scenarios.

7.3.2 Limitations of End-to-End QoE Management considerations

In this thesis, the algorithms implemented consider to capture the QoE-values based on the measurements collected from the networks (packet loss, delay and jitter) which are mapped to QoE metric. The performance of the proposed algorithms is based on the QoE

values collected at the application level (initial delay, stalling events, stall duration and number of bitrate switching. It is important to mention that, QoE control and management should be considered in an end-to-end manner by investigating and collecting parameters at the server, network and application or user's terminals side using SAND technology. Thanks to SAND development where key video quality parameters from both the network and the client side can be collected using a centralized node. The proposed algorithms are limited either to the network side or application side and not both. This makes these algorithms unable to allow DASH clients to send information to the network in real-time and therefore making clients to only depends on the conditions available from the network side. It is worth noting that, DASH is a client-driven adaptation strategy that does not consider end-to-end management aspects for multimedia services delivery.

7.3.3 Limited considerations of Different Video Content Types

The underwhelming growth of UHD has opened up the opportunity for OTT services to promote 4K/8K video streaming services as premium, adding a differentiating edge to consumers. For example, Netflix provides OTT services with UHD contents such as Amazon Prime Video, iTunes, Vimeo, Vudu and YouTube [209]. In this work, only three types of video contents categorized into fast, slow and medium movements are used for experiments as described in Chapters 4 to 6. For example, the BasketballDrive, BQTerrace and ParkScene, Vidyo and the Big Buck Bunny video sequences are used with limited video resolutions ranging from 360p, 480p, 720p and 1080p. However, emerging video streaming services with higher video resolutions such as 4K/8K/12K were not considered in this work. There is a need for the proposed algorithms to be tested using these emerging video services with higher resolutions.

7.3.4 Limited considerations of Different Video Streaming Patterns

This work is mostly limited to a centralized controller where in real-world scenario, more than one controller can be implemented to provide control to more than one network elements in form of clusters. Such design could avoid a single point of failure such that when one controller fails, then the network control can be moved to another OpenDaylight controller. The proposed algorithms can be extended to run in a distributed environments where video streaming sessions are controller by more than on controller in the SDN

network.

7.4 Promising Directions of Future work in QoE Management

The implemented algorithms and architectures can be extended and overcome some of QoE challenges in different research areas of video streaming and the general management and orchestration of resources in future softwarized 5G networks. However, as we move towards future 5G networks, there are significant research areas where this work can be incorporated. In that aspect, this section, presents future research directions and recommendations to replicate this work in the area of QoE-oriented network sharing and slicing, emerging multimedia services and applications in 5G networks and QoE management and orchestration of resources in future 5G softwarized systems that leverage the integration of SDN and NFV.

7.4.1 QoE-oriented Network Sharing and Slicing in Future Softwarized Networks

Moving from hardware-based to software-based platforms could potentially simplify the multi-tenancy support where multiple services/applications from different vertical-specific use cases can be accommodated over a common SDN/NFV-based infrastructure. Besides, evolving the network sharing paradigm to the concept of network slicing that enables multiple VNFs to be configured on the same NFV platform results in many network slice management problems. Although the dynamic resource sharing among slice tenants would make network resource utilization more efficient, the contribution of work can be extended towards implementing an intelligent scheduling algorithms that may allocate resources among these slices in 5G softwarized networks. Besides, the problems concerning NFs placement within a slice, intra/inter-slice QoE management still needs significant efforts to achieve and realize the effectiveness of the network slicing concept in future networks. The QoE-aware MPTCP/SR architecture presented in section 6.6 can be extended towards this direction.

Also, another research direction that needs extensive exploration is related to the isolation between slices, mobility management, dynamic slice creation, and security. Concerning isolation, a set of consistent policies and appropriate mechanisms have to be clearly

defined at each virtualization layer. On the other hand, regarding performance, specific service performance and QoE requirements have to be met on each slice, regardless of network congestion and performance levels of other slices. With security and privacy, efficient mechanisms have to be developed to ensure that any attacks or faults occurring in one slice must not have an impact on another slice. That way, the network sharing and slicing in future softwarized networks using SDN and NFV can be realized in a practical implementation [210].

7.4.2 Emerging Multimedia Services and Applications in 5G Networks

The recent developments in the field of video streaming such as the introduction of higher resolution videos (4K/8K) and newer video formats such as High Dynamic Range (HDR)/Wide Color Gamut (WCG), Light Field, etc. along with the growing number of video-capable devices are the key driving factors for developing intelligent QoE-based management strategies for future video services over softwarized infrastructures. As described in Section 2.5.2, to improve the overall QoE of the users, multimedia communication resources have to be controlled intelligently to ensure as high QoE as possible as done in [29], [30], [12], [101]. Despite these efforts, the current QoE models and algorithms for delivery of adaptive video streams have three limitations: (1) they are developed to capture the behavior of the "average" user, and hence some of them are not personalized, (2) they do not consider the context in which the streaming session takes place, and (3) only the QoE model of the users is inserted into the control loop, but not the user itself. It is worth mentioning that using the network, application and user-level parameters this work could potentially be extended to allow creating QoE personalized models of emerging video streaming services such as 4K/8K or 12K in the future 5G networks. The QoE-fair maximization of multiple competing clients in a shared softwarized environment is another area where this work can be replicated to provide significant improvements of end-user's QoE.

Future 5G networks will support the integration of large IoT devices and applications such as healthcare, smart grids, smart cities, and smart offices as well as transport and smart cars. AR/VR services are a rapidly growing market driven by advances in the device capabilities, consumer excitement about new user experiences, and a range of practical applications (healthcare, automation, etc.). New typical QoE-based models

tailored for these applications have to be developed before their deployment in future networks. For example, given the limited computational performance of small mobile devices and the presence of considerable delays in network communication, many systems (e.g., driver-assistance systems, medicine) are currently restricted to the display of a small amount of mostly static information. As a consequence, these systems are at this stage only of value to a limited set of applications. The QoE management architectures and algorithms developed in this work can be extended to support other services such AR/VR and multimedia IoTs.

7.4.3 QoE Management and Orchestration of Resources in Future 5G Softwarized Networks

Moving legacy NFs from a hardware-centric to software-centric approach using SDN and NFV in future networks not only demands changes on how networks are deployed, operated and managed but also on the orchestration of resources while making sure that the network functions are instantiated in a systematic and on-demand basis. Towards this direction, the ETSI MANO framework has already shown a direction, with anticipated capabilities of life-cycle management and configuration of VNFs. Following that trend, other proposals have appeared that provide solutions for a management platform for VNFs such as Cloud4NFV [211], or NetFATE [212] (which considers the desired QoE of traffic flows during the orchestration of virtualized functions). Relying on the ETSI NFV framework, the AT&T's ECOMP project [213], the Open Source MANO (OSM) project [197], and the ONAP project [214] implement the Service Orchestrator (SO) on top of NFVO. ONAP provides a vendor-agnostic, policy-driven service design, implementation, analytics and life-cycle management for large-scale workloads and services, such as residential vCPE. With ONAP, operators can orchestrate both physical and virtual NFs synchronously. The OPNFV [215] creates a reference NFV platform to accelerate the transformation of enterprise and service provider networks. Still, the OSM [197] from ETSI NFV working group is working on a reference framework that implements MANO functionalities by integrating three other open source platforms (OpenMANO [216], RIFT.ware [217], and JUJU [218]) into a single platform. Other related MANO frameworks and architectures that consider the management and orchestration of both virtualized and non-virtualized functions have been proposed in [22]. Despite these efforts, current proposals are only focused on NFV management and orchestration. There are no efforts given on the management and

orchestration of both SDN and NFV resources in future 5G softwarized networks. While existing projects are focusing on novel architectures that provide the needed flexibility and programmable networks using SDN/NFV or ICN [219], [220] QoE-aware/driven management schemes for multimedia delivery services over future softwarized infrastructures using ML/AI are not covered yet. The QoE-aware MPTCP/SR -enabled architecture for softwarized networks presented in chapter 6 can be extended towards this direction. With the integration of SDN and NFV as shown in Fig. 6.7, the proposed architecture can be extended to support E2E multi-domain management and orchestration of resources across multiple administrative domains in 5G sliced networks. The future implementation of multi-domain in 5G networks would enable the interaction of multiple administrative domains at different levels with different service and infrastructure providers. It would also ensure that video service requests from different domains are mapped into multi-operator and multi-technology domains while matching each service ELA requirements.

7.5 Conclusions

The increasing numbers of data consuming devices which run enhanced applications such as live video streaming, social networking, 3D/HD video, video gaming pose a challenge to mobile and service providers in terms of management of multimedia networks as end-users become accustomed to more resource demanding services with better quality. Fixed and mobile networks today are converging towards the future softwarized networks that will leverage existing and new cutting edge technologies such as SDN and NFV. This integration is set to provide access to any service with better quality to the end-users through a reliable and cost effective communication, over any medium and across multi-operator domains using different networking technologies. In order to meet these requirements, new architectures and intelligent schemes for QoE control and management of future multimedia services have to be designed and developed. As an efforts to respond towards this need, the work in this thesis investigates and develops QoE-centric control and management schemes of video streaming services in software defined and virtualized networks.

The work first presents the background of HAS solutions as the dominant techniques for streaming videos over the best-effort Internet. The work presents next network softwarization and virtualization using SDN and NFV as an important elements for QoE control and

management of multimedia services. This is followed by a description of the proposed QoE-sotwarization aspects of multimedia services using SDN and NFV. The work further provides state-of-the-art solutions and challenges regarding QoE management for HTTP adaptive video streaming using SDN/NFV.

As an important approach to this work, a video quality management scheme based on the traffic intensity under Dynamic Adaptive Video Steaming over HTTP (DASH) using SDN is presented followed by QoE-driven SDN based resource allocation mechanisms for improving the end-users' video quality. In order to meet the aspects of QoE control and management in future networks, this work presents a novel QoE-aware, SDN-based approach for managing multimedia services quality using MPTCP and SR paradigms. MPTCP and SR are utilized to reduce the deployment cost, increase flexibility, reliability, scalability and achieve an optimized E2E QoE level for the end-users in softwarized network. The extensive simulations performed in this work, quality evaluations based on data collected from from real-time multimedia transmissions over the developed QoE-aware SDN/NFV prototype demonstrates the credibility and innovativeness of the new developed QoE control and management schemes for future multimedia services.

Acronyms

BALIA Balanced Linked Adaptation. 79

CAPEX Capital Expenditure. 27

D2D Device to Device. 1

DANE DASH-Aware Network Elements. 19

DASH Dynamic Adaptive Streaming over HTTP. 16

DPI Deep Packet Inspection. 27

ELAs Experience Level Agreements. 4

ETSI European Telecommunication Standard Institute. 27

HAS HTTP Adaptive Streaming. 10

KPI Key Performance Indicators. 100

KQI Key Quality Indicators. 14

LIA Linked Increases Algorithm. 79

M2M Machine to Machine. 1

MANO NFV Management and Orchestration. 29

MCSPM Multi-flow commodity and Constrained Shortest Path Model. 7

MEC Multi-access Edge Computing. 3

MPD Media Presentation Description. 17

NFV Network Function Virtualization. 3

OLIA Opportunistic Linked Increases Algorithm. 79

OPEX Operational Expenditure. 27

OTT Over-The-Top. 16, 44

PED Parameters Enhancing Delivery. 19

PER Parameters Enhancing Reception. 19

QoE Quality of Experience. 1

QoS Quality of Service. 14

QUIC Quick UDP Internet Connections. 16

SAND Server and Network Assisted DASH. 10

SDN Software Defined Networking. 3

SFC Service Function Chaining. 11

SP Service Providers. 2

TCAM Ternary Content-Addressable Memory. 81

TE Traffic Engineering. 7

VNF Virtual Network Function. 100

Bibliography

[1] "ISO/IEC 23009-5: Information Technology - Dynamic Adaptive Streaming over HTTP (DASH) - Part 5: Server and Network Assisted DASH (SAND). Standard," Feb 2017.

[2] G. Cofano, L. D. Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo., "Design and Experimental Evaluation of Network-assisted Strategies for HTTP Adaptive Streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 35, no. 11, pp. 1–24, Nov 2017.

[3] B. Abdelhak, C. B. Ali, Z. Roger, and S. Harous, "SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, Oct 2017.

[4] "OpenFlow Switch Specification," September 2012, [Online: Accessed 11-November-2018]. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.3.1.pdf

[5] M. Li, A. Lukyanenko, Z. Ou, S. Tarkoma, M. Coudron, , and S. Secci, "Multipath Transmission for the Internet: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2887–2925, 2016.

[6] R. Barik, M. Welzl, S. Ferlin, and O. Alay, "Lisa: A linked slowstart algorithm for mptcp," *IEEE International Conference on Communications (ICC)*, pp. 1–7, 2016.

[7] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017 - 2022 white paper," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html.

[8] R. Stankiewicz and A. Jajszczyk, "A survey of QoE assurance in converged networks," *Turk Journal of Electrical Engineering*, vol. 55, no. 7, pp. 1459–1473, 2011.

[9] S. Barakovi and L. Skorin-Kapov, "Survey and Challenges of QoE Management Issues in Wireless Networks," *J. Computer Networks and Communication*, pp. 1–28, 2013.

[10] D. Rivera and A. Cavalli, "QoE-driven service optimization aware of the business model," *30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 725–730, 2016.

[11] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, First quarter 2015.

[12] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards Network-wide QoE Fairness Using Openflow assisted Adaptive Video Streaming," *in Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking,*, pp. 15–20, Feb 2013.

[13] J. Liu, S. Zhang, N. Kato, H. Ujikawa, and K. Suzuki, "Device-to-device communications for enhancing quality of experience in software defined multi-tier LTE-A networks," *IEEE Network*, vol. 29, no. 4, pp. 46–52, 2015.

[14] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE,*, vol. 103, no. 1, pp. 14–76, Dec 2015.

[15] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, S. Member, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, Sept 2016.

[16] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine,*, vol. 55, no. 4, pp. 54–61, April 2017.

[17] I. F. Akyildiz, S.-C. Lina, and P. Wang, "Wireless software-defined networks (W-SDNs) and network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation," *Computer Networks*, vol. 93, pp. 66–79, Dec 2015.

[18] P. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design Considerations for a 5G Network Architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, Nov 2014.

[19] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Communication Magazine*, vol. 54, no. 4, pp. 84–91, April 2016.

[20] P. Neves, R. Cale, M. R. Costa, C. Parada, B. Parreira, J. Alcaraz-Calero, Q. Wang, J. Nightingale, E. Chirivella-Perez, W. Jiang, H. D. Schotten, K. Koutsopoulos, A. Gavras, and M. J. Barros, "The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm," *International Journal of Distributed Sensor Network*, pp. 1–17, Jan 2016.

[21] Y. Wang, P. Li, L. Jiao, Z. Su, N. Cheng, X. S. Shen, and P. Zhang, "A Data-Driven Architecture for Personalized QoE Management in 5G Wireless Networks," *IEEE Wireless Communications Magazine*, vol. 24, no. 1, pp. 102–110, Nov 2016.

[22] M. Gramaglia, I. Digon, V. Friderikos, D. von Hugo, C. Mannweiler, M. Puente, K. Samdanis, and B. Sayadi, "Flexible connectivity and QoE/QoS management for 5G networks: the 5G NORMA," *IEEE International Conference on Communications Workshops (ICC)*, July 2016.

[23] O. Awobuluyi, J. Nightingale, Q. Wang, and J. M. Alcaraz-Calero, "Video Quality in 5G Networks Context-Aware QoE Management in the SDN Control Plane," *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1657–1662, Oct 2015.

[24] M. Dryjanski, "Towards 5G - research activities in Europe (HORIZON 2020 projects," http://grandmetric.com/blog/2016/03/10/towards-5g-research-activities-in-europe-part-2-h2020-projects/, 2017.

[25] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN-Key Technology Enablers for 5G Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468 –2478, Nov 2017.

[26] M. Is-Haka, A. A. Barakabitze, and S. Lingfen, "Video Quality Management over the

Software Defined Networking," *IEEE International Symposium on Multimedia (ISM)*, pp. 559–564, Dec 2016.

[27] E. Grigoriou, A. A. Barakabitze, A. Luigi, S. Lingfen, and P. Virginia, "An SDN-approach for QoE management of multimedia services using resource allocation," *IEEE International Conference Communications (ICC)*, May 2017.

[28] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol, Oct 2000. RFC 2960," 2007.

[29] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeachor, "A Novel QoE-Centric SDN-based Multipath Routing Approach of Mutimedia Services over 5G Networks," *IEEE International Conference on Communications*, May 2018.

[30] A. A. Barakabitze, I.-H. Mkwawa, L. Sun, and E. Ifeachor, "QualitySDN: Improving Video Quality using MPTCP and Segment Routing in SDN/NFV," *IEEE Conference on Network Softwarization*, May 2018.

[31] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeachor, "Multipath protections and dynamic link recovery in softwarized 5g networks using segment routing," in *IEEE GLOBECOM Workshop on Future Internet Architecture, Technologies and Services for 2030 and Beyond*. IEEE, December 2019.

[32] P. L. Callet, S. Möller, and A. Perkis, "Qualinet White Paper on Definitions of Quality of Experience (2012)," *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, vol. 1.2, March 2012.

[33] P. ITU-T Recommendation, "ITU-T Rec. P.10/G.100: Vocabulary for performance and quality of service. Amendment 5: New definitions for inclusion in Recommendation ITU-T P.10/G.100," July 2016.

[34] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE management for cloud applications," *IEEE Communication Magazine*, vol. 50, no. 4, pp. 28–36, Jan 2012.

[35] C. Tselios and G. Tsolis, "On QoE-awareness through Virtualized Probes in 5G Networks," *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, pp. 159–164, Oct 2016.

[36] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 401–418, Firstquarter 2016.

[37] W. Robitza, A. Ahmad, P. A. Kara, L. Atzori, M. G. Martini, A. Raake, and L. Sun, "Challenges of future multimedia QoE monitoring for internet service providers," *Multimedia Tools and Applications*, pp. 1–24, June 2017.

[38] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *INFOCOM, 2014 Proceedings IEEE*, 2014, pp. 82–90.

[39] M. A. Hoque, M. Siekkinen, and J. K. Nurminen., "Energy efficient multimedia streaming to mobile devices: A survey," *IEEE Communication Surveys and Tutorials*, vol. 16, no. 1, pp. 579–597, May 2014.

[40] L. Tao, Y. Gong, S. Jin, and Z. Quan., "Energy efficient video QoE optimization for dynamic adaptive HTTP streaming over wireless networks," *IEEE International Conference on Communication Systems*, pp. 1–6, May 2016.

[41] N. Bouten, R. de O. Schmidt, J. Famaey, S. Latre, A. Pras, and F. D. Turck, "QoE-driven in-network optimization for Adaptive Video Streaming based on packet sampling measurements," *Computer Networks*, vol. 81, pp. 96–115, April 2015.

[42] N. Bouten, J. Famaey, and S. Latre, "QoE optimization through in-network quality adaptation for HTTP Adaptive Streaming," June 2012.

[43] I. Triki, R. El-Azouzi, M. Haddad, Q. Zhu, and Z. Xu, "Learning from experience: A dynamic closed-loop QoE optimization for video adaptation and delivery," pp. 1–5, Oct 2017.

[44] Y. Xu, E. Altman, R. El-Azouzi, M. Haddad, S. Elayoubi, and T. Jimenez, "Analysis of Buffer Starvation With Application to Objective QoE Optimization of Streaming Services," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 813–827, April 2014.

[45] "Global Internet Phenomena Report: North America and Latin America," https://www.sandvine.com/hubfs/downloads/archive/2016-global-internet-phenomena-report-latin-america-and-north-america.pdf, 2016, [Online: Accessed 06-December-2018].

[46] B. Hayes, Y. Chang, and G. Riley, "Omnidirectional Adaptive Bitrate Media Delivery Using MPTCP/QUIC over an SDN Architecture," *IEEE Global Communications Conference*, Dec 2017.

[47] ISO/IEC 23009-1:2014, "Preview Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," https://www.iso.org/standard/65274.html, 2017, [Online: Accessed 06-December-2018].

[48] E. Thomas, M. O. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG DASH Performance via Server and Network Assistance," *SMPTE Motion Imaging Journal*, vol. 126, no. 1, pp. 22–27, Jan 2017.

[49] Microsoft, "Microsoft Silverlight Smooth Streaming," https://www.microsoft.com/silverlight/smoothstreaming/, 2017, [Online: Accessed 06-December-2018].

[50] Apple, "Apple HTTP Live Streaming," https://developer.apple.com/streaming/, 2017, [Online: Accessed 06-December-2018].

[51] Adobe, "Adobe HTTP Dynamic Streaming (HDS)," https://www.adobe.com/devnet/hds.html, 2017, [Online: Accessed 06-December-2018].

[52] "ETSI TS 126 247 V14.1.0 (2017): Universal Mobile Telecommunications System (UMTS); LTE;Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (3GPP TS 26.247 version 14.1.0 Release 14)," http://www.etsi.org/deliver/etsi_ts/126200_126299/126247/14.01.00_60/ts_126247v140100p.pdf/, 2017, [Online: Accessed 10-December-2018].

[53] "GPAC:Multimedia Open Source Project,," [Availableat]:https://gpac.wp.imt.fr/mp4box/.

[54] L. Hung, H. N. Nguyen, and N. P. Ngoc, "Quality-driven bitrate adaptation method for HTTP live-streaming," *IEEE International Conference on Communication Workshop (ICCW), London*, pp. 1771–1776, June 2015.

[55] "Dash-Industry-Forum, dash.js," [Availableat]:https://github.com/Dash-Industry-Forum/dash.js/wiki.

[56] S. Michael, E. Sebastian, S. Martin, Z. Thomas, H. Tobias, and T.-G. Phuoc, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communication Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, Dec 2015.

[57] Netflix, "Per-Title Encode Optimization," https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2, Dec 2015, [Online: Accessed 06-December-2018].

[58] D. Z. Rodríguez, Z. Wang, R. L. Rosa, and G. Bressan, "The impact of video-quality-level switching on user quality of experience in dynamic adaptive streaming over HTTP," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, 2014.

[59] D. Bhat, A. Rizk, M. Zink, and R. Steinmetz, "Network Assisted Content Distribution for Adaptive Bitrate Video Streaming," *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 62–75, June 2017.

[60] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing & softwarization: A survey on principles, enabling technologies & solutions," *IEEE Surveys and Tutorials*, vol. 20, no. 3, pp. 2429–2453, March 2018.

[61] P. A. Frangoudis, L. Yala, A. Ksentini, and T. Taleb, "An architecture for on-demand service deployment over a Telco CDN," pp. 1–6, May 2016.

[62] S. Retal, M. Bagaa, T. Taleb, and H. Flinck, "Content delivery network slicing: QoE and cost awareness," *IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.

[63] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeachor, "A Novel QoE-Aware SDN-enabled, NFV-based Management Architecture for Future Multimedia Applications on 5G Systems," in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, Lisbon, Portugal, 2016.

[64] "Open Networking Foundation (ONF)," https://www.opennetworking.org/, [Online: Accessed 06-December-2018].

[65] M. Smith, M. Dvorkin, Y. Laribi, V. Pandey, P. Garg, and N. Weidenbache, "OpFlex Control Protocol, Internet Draft, Internet Engineering Task Force,"

April 2014, [Online: Accessed 11-December-2018]. [Online]. Available: http://tools.ietf.org/html/draft-smith-opflex-00

[66] A. Doria, J. H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern, "Internet Eng. Task Force (IETF): Forwarding and Control Element Separation (ForCES)," http://tools.ietf.org/html/draft-smith-opflex-00, Fremont, CA, USA, March 2010, [Online: Accessed 13-December-2018].

[67] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: A survey ," *IEEE Communication Surveys Tutorials*, vol. 16, no. 1, pp. 493–512, 2014.

[68] M. Taylor, "A guide to NFV and SDN. White paper, Metaswitch Networking, London, U.K., Dec. 2014," http://www.metaswitch.com/sit,es/default/files/Metaswitch_WhitePaper_NFVSDN_final_rs.pdf, [Online: Accessed 12-December-2018].

[69] "The VMware NSX network virtualization platform. Technical white paper," https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/products/nsx/vmware-nsx-network-virtualization-platform-white-paper.pdf, [Online: Accessed 06-December-2018].

[70] H. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software defined networking," *IEEE Communications. Magazine*, vol. 51, no. 2, pp. 136–141, May 2013.

[71] "Floodlight is a Java-based OpenFlow controller," http://floodlight.openflowhub.org/, 2012, [Online: Accessed 03-December-2018].

[72] D. Erickson, "The Beacon Openflow Controller," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 13–18.

[73] Nippon Telegraph and Telephone Corporation, "Ryu Network Operating System, 2012," http://osrg.github.com/ryu/, [Online: Accessed 16-December-2018].

[74] "The POX network software platform," https://github.com/noxrepo/pox/, [Online: accessed 27-February-2019].

[75] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for openflow," in *Proceedings of the 2010 Internet Network Management Conference on Research*

on Enterprise Networking, ser. INM/WREN'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 3–3.

[76] HP, "HP SDN controller architecture," http://h17007.www1.hpe.com/docs/networking/solutions/sdn/devcenter/06_-_HP_SDN_Controller_Architecture_TSG_v1_3013-10-01.pdf, 2013, [Online: Accessed 06-December-2018].

[77] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed multi-domain SDN controllers," in *IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–4.

[78] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "Onos: Towards an open, distributed sdn os," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN'14. New York, NY, USA: ACM, 2014, pp. 1–6.

[79] O. Yustus Eko, L. SangGon, L. HoonJae, and L. JunHuy, "Distributed SDN controller system," *Journal Computer Networks:*, vol. 121, no. C, pp. 100–111, July 2017.

[80] A. Malishevskiy, D. Gurkan, L. Dane, R. Narisetty, S. Narayan, and S. Bailey, "OpenFlow-Based Network Management with Visualization of Managed Elements," *Third GENI Research and Educational Experiment Workshop*, pp. 73–74, 2014.

[81] J. Halpern and C. Pignataro, "IETF:Service Function Chaining (SFC) Architecture," https://tools.ietf.org/html/rfc7665, Oct 2015.

[82] ETSI, "ETSI GS NFV-MAN 001 V1.1.1: Network Functions Virtualisation (NFV); Management and orchestration, ETSI Ind. Spec. Group (ISG) Netw. Functions Virtualisation (NFV)," Sophia-Antipolis Cedex, France, pp. 1–183, Dec 2014.

[83] ——, "ETSI GS NFV V1.1.1: Network Functions Virtualisation (NFV); Architectural Framework," http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf, Sophia-Antipolis Cedex, France, Oct 2013, [Online: Accessed 06-December-2018].

[84] "ETSI GS NFV 003 V1.2.1: Network Functions Virtualisation (NFV);Terminology for main concepts in NFV, ETSI Ind. Spec. Group (ISG) Netw. Functions Virtualisation (NFV)," http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_

60/gs_nfv-man001v010101p.pdf, Sophia-Antipolis Cedex, France, Dec 2014, [Online: Accessed 06-December-2018].

[85] "ETSI GS NFV-PER 002 V1.1.2: Network Functions Virtualisation (NFV); Proof of Concepts; Framework," http://www.etsi.org/deliver/etsi_gs/NFV-PER/001_099/002/01.01.02_60/gs_NFV-PER002v010102p.pdf, Sophia-Antipolis Cedex, France, Dec 2014, [Online: Accessed 06-December-2018].

[86] H. S, G. J, and L. K, "Network Hypervisors: Enhancing SDN Infrastructure," *Computer Communications*, vol. 46, pp. 87–96, 2014.

[87] A. Blenk, A. Basta, J. Zerwas, M. Reisslein, and W. Kellerer, "Control Plane Latency with SDN Network Hypervisors: The Cost of Virtualization," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 366–380, Sept 2016.

[88] J. Smith and R. Nair, "The Architecture of Virtual Machines, Computer," *Computer Network*, vol. 5, no. 3, pp. 32–38, May 2005.

[89] I. Afolabi, M. Bagaa, T. Taleb, and H. Flinck, "End-to-End Network Slicing Enabled Through Network Function Virtualization," *IEEE Conference on Standards for Communications and Networking (CSCN)*, no. 5, pp. 30–35, Sept 2017.

[90] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An Updated Performance Comparison of Virtual Machines and Linux Containers," *IBM Research Report*, July 2014.

[91] G. X. Miguel, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange, and C. A. F. D. Rose, "Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments," *21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pp. 233–240, Feb 2013.

[92] C. Boettiger, "Introduction to docker for reproducible research," *SIGOPS Oper. Syst. Rev*, vol. 49, no. 1, pp. 71–79, Jan 2013.

[93] B. des Ligneris, "Virtualization of Linux based computers: the Linux-VServer project," *High Performance Computing Systems and Applications, 19th International Symposium on*, June 2005.

[94] "OpenVz: Open source container-based virtualization for Linux." https://openvz.org/, [Online: accessed 20-October-2018].

[95] A. Barakabitze, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. Martini, and L. Atzori, "QoE Management of Multimedia Services in Future Networks: A Tutorial and Survey," *IEEE Communication Surveys and Tutorials*, pp. 1–1, December 2019.

[96] S. Yusuf, M. Andreas, and E. Christopher, "Adaptive Bitrate Selection: A Survey," *IEEE Communication Surveys & Tutorials*, vol. 19, no. 4, pp. 2985–3014, July 2017.

[97] M. Xiao, V. Swaminathan, S. Wei, and S. Chen, "DASH2M: Exploring HTTP/2 for Internet Streaming to Mobile Devices," *In Proceedings of the ACM on Multimedia Conference*, pp. 22–31, 2016.

[98] M. Mu, M. Broadbent, A. Farshad, N. Hart, D. Hutchison, Q. Ni, and N. Race, "A Scalable User Fairness Model for Adaptive Video Streaming Over SDN-Assisted Future Networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2168–2184, Aug 2016.

[99] A. Bentaleb, A. C. Begen, and R. Zimmermann, "QoE-Aware Bandwidth Broker for HTTP Adaptive Streaming Flows in an SDN-Enabled HFC Network," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 575–589, June 2018.

[100] L. Eirini, S. Konstantinos, P. Emmanouil, P. Nikos, and M. Lazaros, "QoE-SDN APP: A Rate-guided QoE-aware SDN-APP for HTTP Adaptive Video Streaming," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–1, July 2018.

[101] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH : Improving QoE of HTTP adaptive streaming using software defined networking," *in Proc. ACM Conference on Multimedia*, pp. 1296–1305, Oct 2016.

[102] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An SDN architecture with dash assisting network elements," *in 7th International ACM Conference on Multimedia Systems*,, May 2016.

[103] A. Mansy, M. Fayed, and M. Ammar., "Network-Layer Fairness for Adaptive Video Streams," *IFIP Networking Conference (IFIP Networking)*, pp. 1–9, May 2015.

[104] A. D. Ferguson, A. Guha, C. L. R. Fonseca, and S. Krishnamurthi, "Participatory Networking: An API for Application Control of SDNs," *In Proc. ACM SIGCOMM 2013*,, pp. 327–338, Aug 2013.

[105] J. Chen, M. Ammar, M. Fayed, and R. Fonseca, "Client-Driven Network level QoE Fairness for Encrypted 'DASH-S'," *in Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks,*, pp. 55–60, Aug 2016.

[106] S. Ramakrishnan, X. Zhu, F. Chan, K. K. R. Kambhatla, Z. Lu, C. Chan, and B. Krishnamurthy, "Optimizing Quality-of-Experience for HTTP-based Adaptive Video Streaming: An SDN-based Approach," *International Journal of Multimedia Data Engineering and Management*, vol. 7, no. 4, pp. 22–44, Dec 2016.

[107] J. Jiang, L. Hu, P. Hao, R. Sun, J. Hu, and H. Li, "Q-FDBA: improving QoE fairness for video streaming," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10 787–10 806, May 2018.

[108] T. Hoßfeld, L. Skorin-Kapov, P. E. Heegaard, and M. Varela, "Definition of QoE Fairness in Shared Systems," *IEEE Communications Letters,*, vol. 21, no. 1, Jan 2017.

[109] O. Dobrijevic, M. Santl, and M. Matijasevic, "Ant colony optimization for QoE-centric flow routing in software-defined networks," *in Proceedings of the 11th International Conference on Network and Service Management (CNSM)*, pp. 274–278, Nov 2015.

[110] O. Dobrijevic, A. Kassler, L. Skorin-Kapov, and M. Matijasevic, "Q-POINT: QoE-Driven Path Optimization Model for Multimedia Services," *Wired/Wireless Internet Communications*, vol. 8458, pp. 134–147, 2014.

[111] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, H. M. Sparks, R., and E. Schooler, "SIP: Session Initiation Protocol. IETF ," *IETF*, 2002.

[112] A. KasslerA, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, and P. Dely, "Towards QoE-driven Multimedia Service Negotiation and Path Optimization with Software Defined Networking," *in 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Nov 2012.

[113] L. Skorin-Kapov, M. Mosmondor, O. Dobrijevic, and M. Matijasevic, "Application-level QoS Negotiation and Signaling for Advanced Multimedia Services in the IMS," *IEEE Comm. Mag,*, vol. 45, no. 7, pp. 108–116, July 2007.

[114] J. Yang, E. Yang, Y. Ran, and S. Chen, " SDM^2Cast:An OpenFlow-Based, Software-

Defined Scalable Multimedia Multicast Streaming Framework," *IEEE Internet Computing*, vol. 19, no. 4, pp. 36–44, June 2015.

[115] K. A. Noghani and M. O. Sunay, "Streaming Multicast Video over Software-Defined Networks," *IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2014.

[116] A. Dimitris, P. Ilias, L. Asimakis, T. Christos, and D. Tasos, "End-to-end quality aware optimization for multimedia clouds," *International Conference on Telecommunications and Multimedia (TEMU)*, pp. 1–5, July 2016.

[117] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware Video Streaming using SDN," *In Proceedings of IEEE Global Communications Conference(GLOBECOM),*, pp. 1317–1322, February 2014.

[118] R. I. T. da Costa Filho, W. Lautenschläger, N. Kagami, M. C. Luizelli, V. Roesler, and L. P. Gaspary, "Scalable QoE-aware Path Selection in SDN-based Mobile Networks," in *2018 IEEE International Conference on Computer Communications (INFOCOM)*, April 2018, pp. 1–6.

[119] R. I. T. da Costa Filho, W. Lautenschläger, N. Kagami, V. Roesler, and L. P. Gaspary, "Network Fortune Cookie: Using Network Measurements to Predict Video Streaming Performance and QoE," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[120] C. Cetinkaya, E. Karayer, M. Sayit, and C. Hellge, "SDN for Segment Based Flow Routing of DASH," *IEEE Fourth International Conference on Consumer Electronics*, pp. 74–77, June 2014.

[121] N. Bouten, M. Claeys, B. Van Poecke, S. Latré, and F. De Turck, "Dynamic Server Selection Strategy for Multi-Server HTTP Adaptive Streaming Services," *12th International Conference on Network and Service Management (CNSM)*, pp. 82–90, Nov 2016.

[122] C. Wang, H. Kim, and R. Morla, "QoE Driven Server Selection for VoD in the Cloud," in *2015 IEEE 8th International Conference on Cloud Computing*, June 2015, pp. 917–924.

[123] U. Tuba, C. Cihat, and S. Muge, "Learning-based approach for layered adaptive video streaming over SDN," *Computer Networks*, vol. 92, pp. 357–368, Dec 2015.

[124] G. Ankit, G. Megha, S. G. Manoj, L. Vijay, and M. Conti, "ELBA: Efficient Layer Based routing Algorithm in SDN," *25th International Conference on Computer Communication and Networks (ICCCN)*, Aug 2016.

[125] P. Qin, B. Dai, B. Huang, and G. Xu, "Bandwidth-aware scheduling with sdn in hadoop: A new trend for big data," *IEEE Systems Journal*, vol. 4, no. 11, pp. 2337–2344, 2015.

[126] W. Zhou, L. Li, M. Luo, and W. Chou, "Rest api design patterns for sdn northbound api," *28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 358–365, May 2014.

[127] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying qoe optimal adaptation of http adaptive streaming based on subjective studies," *Computer Networks*, vol. 81, no. 11, pp. 320–332, April 2015.

[128] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," *ACM conference on Internet measurement conference*, pp. 225–238, November 2012.

[129] D. Palma, J. Goncalves, B. Sousa, L. Cordeiro, P. Simoes, S. Sharma, and D. Staessens, "The queuepusher: Enabling queue management in openflow," *Third European Workshop on Software-Defined Networks*, pp. 125–126, Sept 2014.

[130] M. Jarschel, S. Oechsner, D. Schlosser, S. G. R. Pries, and P. Tran-Gia, "Modeling and performance evaluation of an openflow architecture," *23rd international teletraffic congress. International Teletraffic Congress*, pp. 1–7, Sept 2011.

[131] E. Liotou, G. Tseliou, K. Samdanis, F. A. D. Tsolkas, and C. Verikoukis, "An sdn qoe-service for dynamically enhancing the performance of ott applications," *Seventh International Workshop in Quality of Multimedia Experience (QoMEX)*, pp. 1–2, May 2015.

[132] J. Ansell, W. K. G. Seah, B. Ng, and S. Marshall, "Making queueing theory more palatable to sdn/openflow-based network practitioners," *IEEE/IFIP Network Operations and Management Symposium*, pp. 1119–1124, April 2016.

[133] J. Hu, C. Lin, X. Li, and J. Huang, "Scalability of control planes for software defined

networks: Modeling and evaluation," *in IEEE 22nd International Symposium of Quality of Service*, pp. 147–152, May 2014.

[134] F. Kelly, "Charging and rate control for elastic traffic," vol. 8, pp. 33–37, May 1997.

[135] V. Tsibonis, L. Georgiadis, and L. Tassiulas, "Exploiting wireless channel State information for throughput maximization," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2566–2582, May 2004.

[136] R. M. Abuteir, A. Fladenmuller, and O. Fourmaux, "SDN Based Architecture to Improve Video Streaming in Home Networks," *IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 220–226, 2016.

[137] R. F. Diorio and V. S. Timoteo, "Multimedia Content Delivery in OpenFlow SDN: An Approach Based on a Multimedia Gateway," *International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec 2016.

[138] Y. Jin, Y. Wen, and C. Westphal, "Towards joint resource allocation and routing to optimize video distribution over future internet," *IFIP Networking Conference (IFIP Networking)*, May 2015.

[139] K. T. Bagci, K. E. Sahin, and A. M. Tekalp, "Queue-allocation optimization for adaptive video streaming over software defined networks with multiple service-levels," *IEEE International Conference on Image Processing (ICIP)*, Sept 2016.

[140] T. Hoßfeld, D. Hausheer, F. Hecht, F. Lehrieder, S. Oechsner, I. Papafili, P. Racz, S. Soursos, D. Staehle, G. D. Stamoulis, P. Tran-Gia, and B. Stille, "An Economic Traffic Management Approach to Enable the TripleWin for Users, ISPs, and Overlay Providers," *FIA Prague Book. Towards the Future Internet âĂŤ A European Research Perspective: IOS Press Books Online*, May 2009.

[141] L. Skorin-Kapov, K. Ivesic, G. Aristomenopoulos, and S. Papavassiliou, "Approaches for utility-based QoE-driven optimization of network resource allocation for multimedia services," *Data traffic monitoring and analysis*, pp. 337–358, Feb 2013.

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

[143] H. J. Kim, D. G. Yun, H. S. Kim, K. S. Cho, and S. G. Choi, "QoE assessment model for video streaming service using QoS parameters in wired-wireless network," *14th International Conference on Advanced Communication Technology (ICACT)*, pp. 459–464, Feb 2012.

[144] M. R. Celenlioglu, S. B. Goger, and H. A. Mantar, "An SDN-based energy-aware routing model for intra-domain networks," *Software, Telecommunications and Computer Networks (SoftCOM), 22nd International Conference on, Split*, pp. 61–66, 2014.

[145] "Mininet: An Instant Virtual Network on your Laptop (or other PC," http://mininet.org/, 2019, [Online: Accessed 06-June-2019].

[146] "OpenDaylight Platform," https://www.opendaylight.org/, 2019, [Online: Accessed 06-June-2019].

[147] "Video LAN," http://www.videolan.org, 2019, [Online: Accessed 06-May-2019].

[148] "Emulating the delay, packet loss with NetEM," https://wiki.linuxfoundation.org/networking/netem, 2019, [Online: Accessed 06-May-2019].

[149] M. Mu, M. Broadbent, A. Farshad, N. Hart, D. Hutchison, Q. Ni, and N. Race, "Managing quality of experience for wireless VoIP using noncooperative games ," in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, February 2016, pp. 1193–1204.

[150] M. Agiwal, A. Roy, and N. Saxena, "Next Generation 5G Wireless Networks:A Comprehensive Survey," *IEEE Communication Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[151] F. B. Osseiran, V. Braun, K. Kusume, M. M. P. Marsch, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, B. T. M. Uusitalo, and M. Fallgren, "Scenarios for 5G mobile and wireless communications: The vision of the METIS project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, May 2014.

[152] S. Chen and J. Zhao, "The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 36–43, May 2014.

[153] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The Segment Routing Architecture," *IEEE Global Communications Conference (GLOBECOM)*, Feb 2015.

[154] M. Sandri, A. Silva, L. A. Rocha, and F. L. Verdi, "On the Benefits of Using Multipath TCP and Openflow in Shared Bottlenecks," *IEEE 29th International Conference on Advanced Information Networking and Applications*, April 2015.

[155] S. Tariq and M. Bassiouni, "Packet-In Message Control for Reducing CPU Load and Control Traffic in OpenFlow Switches," *In Proceedings of IEEE CCNC*, January 2015.

[156] S. Balazs, N. Felician, C. Levente, L. Gulyas, and A. Gulyas, "SDN based Testbeds for Evaluating and Promoting Multipath TCP," *In Proceedings of IEEE ICC- Next-Generation Networking Symposium*, pp. 3044–3050, June 2014.

[157] S. Zannettou, S. Michael, and P. Fragkiskos, "Exploiting path diversity in datacenters using MPTCP-aware SDN," *IEEE Symposium on Computers and Communication (ISCC),*, pp. 539–546, June 2016.

[158] H. Nam, C. Doru, and S. Henning, "Towards Dynamic MPTCP Path Control Using SDN," *IEEE NetSoft Conference and Workshops (NetSoft),*, pp. 286–294, July 2016.

[159] J. Duan and C. W. Zhi Wangy, "Responsive multipath TCP in SDN-based datacenters," *IEEE International Communication Conference (ICC)*, June 2015.

[160] H. Ali, I. H. Elhajj, C. Ali, and A. Kayssi, "SDN for MPTCP: An Enhanced Architecture for Large Data Transfers in Datacenters," *IEEE International Communication Conference (ICC)*, May 2017.

[161] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Streaming High-Quality Mobile Video with Multipath TCP in Heterogeneous Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2345–2361, March 2016.

[162] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon, "Cross-layer scheduler for video streaming over MPTCP,," *Proceedings of the 7th International Conference on Multimedia Systems*, May 2016.

[163] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, and P.Castoldi, "Segment Routing for Effective Recovery and Multi-domain Traffic Engineering," *Journal of Optical Communication Networks*, vol. 9, pp. A223–A232, February 2017.

[164] D. Luca, V. Luca, V. Pier Luigi, S. Giuseppe, and S. Stefano, "Traffic Engineering with Segment Routing: SDN-based Architectural Design and Open Source Implementation," *Fourth European Workshop on Software Defined Networks (EWSDN)*, Sept 2015.

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

[166] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeachor, "A Novel QoE-Centric SDN-based Multipath Routing Approach of Mutimedia Services over 5G Networks," *IEEE International Conference on Communications*, May 2018.

[167] Y. Liu, Y. Li, M. Caniniy, Y. Wang, and J. Yuan, "Scheduling multi-flow network updates in Software-Defined NFV systems," *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016.

[168] A. Md. Abbas and B. Alcardo Alex, "Evolution of LTE and Related Technologies towards IMT-Advanced," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 1, pp. 16–22, January 2015.

[169] B. Alcardo Alex and X. Tan, "Performance Analysis of PAPR Reduction Using DFT-Spreading OFDMA Technique," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 3, pp. 1401–1409, March 2015.

[170] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of Multipath TCP schedulers," *In ACM SIGCOMM Workshop on Capacity Sharing*, vol. 11, pp. 27–32, Aug 2014.

[171] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, , and M. Handley, "How hard can it be? Designing and implementing a deployable Multipath TCP," *In USENIX NSDI*, April 2012.

[172] I. Lopez, M. Aguado, C. Pinedo, and E. Jacob., "SCADA systems in the railway domain: Enhancing reliability through redundant multipathtcp," *In IEEE ITSC*, Sept 2015.

[173] K. Argyraki and D. R. Cheriton, "Loose source routing as a mechanism for traffic

policies," *ACM SIGCOMM Future Directions in Network Architecture*, pp. 57–64, August 2004.

[174] C. Raiciu and M. H. D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," *IETF, Internet Standart RFC 6356*, Oct 2011.

[175] R. Khalili, N. Gast, and M. Popovic, "Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP," *8th USENIX Symposium on Netwroked System Design and Implementation (NSDI)*, vol. 11, pp. 8–8, Jul 2014.

[176] A. Walid, Q. Peng, J. Hwang, and S. Low, "Balanced Linked Adaptation Congestion Control Algorithm for MPTCP," *IETF, Internet Draft draftwalid-mptcp-congestion-control-04*, Jan 2016.

[177] M. Xu, Y. Cao, , and E. Dong, "Balanced Linked Adaptation Congestion Control Algorithm for MPTCP," *IETF, Internet Draft draftwalid-mptcp-congestion-control-04*, Jul 2016.

[178] N. Kukreja, G. Maier, R. Alvizu, and A. Pattavina, "SDN based Automated Testbed for Evaluating Multipath TCP," *IEEE International Conference on Communications Workshops (ICC)*, pp. 718–723, May 2016.

[179] L. Luo, H. Yu, and S. Luo, "Achieving Fast and Lightweight SDN Updates with Segment Routing," *IEEE Global Communications Conference (GLOBECOM),*, pp. 1–6, Dec 2016.

[180] ITEC, "Dynamic Adaptive Streaming over HTTP:Dash plugin for vlc,," [Availableat]: http://www-itec.uni-klu.ac.at/dash/?page_id=10, June 2012.

[181] T. Mangla, N. Theera-Ampornpunt, M. Ammar, E. Zegura, and S. Bagchi, "Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments," May 2016.

[182] X. K. Zou, J. Erman, V. Gopalakrishnan, E. Halepovic, R. Jana, X. Jin, J. Rexford, and R. K. Sinha, "Can accurate predictions improve video streaming in cellular networks?" pp. 57–62, 2015.

[183] C. Chatfield, "The analysis of time series: an introduction," pp. 57–62, 2004.

[184] A. E. Al-Issa, A.Bentaleb, A. A. Barakabitze, T. Zinner, and B. Ghita, "Bandwidth prediction schemes for defining bitrate levels in sdn enabled adaptive streaming," pp. 57–62, 2015.

[185] S. Jain, A. Kumar, S. Mandal, J.Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined WAN," *Proceedings of ACM SIGCOMM*, pp. 3–14, Aug 2013.

[186] Y. Hong, S. Kandula, R. Mahajan, V. G. M. Zhang, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven WAN," *Proceedings of ACM SIGCOMM*, pp. 15–20, Aug 2013.

[187] G. Sallam, G. R. Gupta, B. Li, and B. Ji, "Shortest path and maximum flow problems under service function chaining constraints," pp. 57–62, 2018.

[188] C. James, E. Halepovic, M. Wang, R. Jana, and N. K. Shankaranarayanan, "Is Multipath TCP (MPTCP) Beneficial for Video Streaming over DASH?" *IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems,*, pp. 331–336, September 2016.

[189] A. AbdelSalam, F. Clad, C. Filsfils, S. Salsano, G. Siracusano, and L. Veltri, "Implementation of virtual network function chaining through segment routing in a linux-based NFV infrastructure," *IEEE Conference on Network Softwarization (NetSoft),*, Aug 2017.

[190] K. D. Joshi and K. Kataoka, "SFO: SubFlow Optimizer for MPTCP in SDN,," *26th IEEE International Telecommunication Networks and Applications Conference (ITNAC) ,*, pp. 173–178, December 2016.

[191] S. Chung, S. Moon, and S. Kim, "The virtualized MPTCP proxy performance in cellular network," *Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 703–707, July 2017.

[192] T.-T.-L. Nguyen, T.-M. Pham, and H. T. T. Binh, "Adaptive multipath routing for network functions virtualization," *Proceedings of the 7th Symposium on Information and Communication Technology*, pp. 222–228, December 2016.

[193] C. Thaenchaikun and W. Panichpattanakul, "Control-plane OpenFlow segment routing SDN for network control messages reduction,," *14th IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* ,, pp. 322–325, June 2017.

[194] G. Trimponias, Y. Xiao, H. Xu, X. Wu, and Y. Geng, "On Traffic Engineering with Segment Routing in SDN based WANs,," *IEEE/ACM Trans. Networking*,, vol. 25, no. 5, pp. 1–17, September 2017.

[195] A. Hirwe and K. Kataoka, "LightChain: A lightweight optimisation of VNF placement for service chaining in NFV ,," *IEEE NetSoft Conference and Workshops (NetSoft)*,, pp. 33–37, June 2016.

[196] E. Datsika, A. Antonopoulos, N. Zorba, and C. Verikoukis, "Software Defined Network Service Chaining for OTT Service Providers in 5G Networks," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 124–131, November 2017.

[197] "OSM ESTI NFV," https://osm.etsi.org/, [Online: Accessed 17-December-2018].

[198] F. Ongaro, E. Cerqueira, L. Foschini, A. Corradi, and M. Gerla, "Enhancing the Quality Level Support for Real-time Multimedia Applications in Software-Defined Networks," *International Conference on Computing, Networking and Communications, Communication QoS and System Modeling Symposium*, pp. 505–509, June 2015.

[199] H. Shen, Y. Liu, T. Wang, H. Yang, and L. Sang, "QoE-optimal rate adaptation for HTTP adaptive streaming," *IEEE/CIC International Conference on Communications in China (ICCC)*, July 2016.

[200] S. Akhshabi, A. Begen, and C.Dovrolis, "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP," *in Proc.of ACM MMSys*, pp. 157–169, 2011.

[201] Y. Liu and S. Dey, "User experience modeling for dash video," *IEEE Packet Video Workshop*, pp. 1–8, July 2013.

[202] Z. Dizhi, S. Wei, and S. Minghui, "Goodput improvement for multipath TCP by congestion window adaptation in multi-radio devices," *IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Jan 2013.

[203] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, "Traffic Engineering with Segment Routing: SDN-Based Architectural Design and Open Source Implementation," *2015 Fourth European Workshop on Software Defined Networks (EWSDN)*, Nov 2015.

[204] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet video delivery in YouTube: from traffic measurements to quality of experience. In: Biersack, E., Callegari, C., Matijasevic, M. (Eds.), Data Traffic Monitoring and Analysis. Springer," pp. 264–301, June 2013.

[205] J. W. Suurballe, "Disjoint paths in a network," *International Journal on Networks*, vol. 4, no. 2, pp. 125–145, 1974.

[206] A. K. Saha, K. Sambyo, and C. Bhunia, "Topology Discovery, Loop Finding and Alternative Path Solution in POX Controller," *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, 2016.

[207] B. J. van Asten, N. L. van Adrichem, and F. A. Kuipers, "Scalability and resilience of software-defined networking: An overview," *arXiv preprint arXiv:1408.6760*, pp. 1–19, Aug 2014.

[208] Z. Liehuang, M. M. Karim, S. Kashif, M. Fan Li, D. Xiaojiang, and M. Guizani, "SDN Controllers: Benchmarking & Performance Evaluation," *T IEEE JSAC*, 2019.

[209] "Technology, Media & Telecom OTT Video And Connected Devices Drive 4K UHD, As 8K Emerges," https://www.spglobal.com/marketintelligence/en/news-insights/research/ott-video-and-connected-devices-drive-4k-uhd-as-8k-emerges, Dec 2018.

[210] A. B. Alcardo, A. Ahmad, R. Mijumbi, and A. Hines, "5G Network Slicing using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges," *Computer Networks*, vol. 167, pp. 1–40, November 2019.

[211] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento, "Cloud4NFV: A platform for virtual network functions," *3rd IEEE International Conference on Cloud Networking*, pp. 288–293, Oct 2014.

[212] V. Riccobene, A. Lombardo, A. Manzalini, and G. Schembra, "Network Functions At The Edge (NetFATE): Design and Implementation Issues," pp. 1–6, 2014.

[213] "AT & T. Enhanced Control, Orchestration, Management, and Policy (ECOMP) Architecture," https://about.att.com/content/dam/snrdocs/ecomp.pdf, [Online: Accessed 17-December-2018].

[214] "ONAP:Linux Foundation Project," https://www.open-o.org/, [Online: Accessed 17-December-2018].

[215] "Open Platform for NFV (OPNFV)," https://www.opnfv.org/about, Jan 2015.

[216] "OpenMANO," https://osm.etsi.org/, [Online: Accessed 12-December-2018].

[217] "RIFT.io," https://riftio.com/tag/rift-ware/, [Online: Accessed 17-December-2018].

[218] "Anssr Data-Engine," http://www.ubuntu.com/cloud/juju, [Online: Accessed 12-December-2018].

[219] A. A. Barakabitze, T. Xiaoheng, and G. Tan, "A Survey on Naming, Name Resolution and Data Routing in Information Centric Networking (ICN)," *Int. J. Adv. Res. Comput. Commun. Eng*, vol. 3, no. 10, pp. 8322–8330, 2014.

[220] A. A. Barakabitze and T. Xiaoheng, "Caching and Data Routing In Information Centric Networking (ICN): The Future Internet Perspective," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 11, pp. 26–36, Jan 2014.