Universita degli Studi di Cagliari

PhD DEGREE

Department of Electronic and Computer Engineering (DRIEI)

Cycle XXXII

TITLE OF THE PHD THESIS

Quality of Experience monitoring and management strategies for future smart networks

Scientific Disciplinary Sector(s)

ING-INF/03

PhD Student: Elisavet Grigoriou

Coordinator of the PhD Programme: Alessandro Giua

Supervisor: Prof.Luigi Atzori

Co-Supervisor: Prof.Virginia Pilloni

Final exam. Academic Year 2018 - 2019

Thesis defence: January - February 2020 Session

Quality of Experience monitoring and management strategies for future smart networks

Elisavet Grigoriou

Supervisor: Prof.Luigi Atzori

Co-Supervisor: Prof.Virginia Pilloni

Department of Electronic and Computer Engineering (DRIEI)

University of Cagliari, Italy

This dissertation is submitted for the degree of

Doctor of Philoshophy

January - February 2020

Acknowledgements

Firstly, I would like to express my special appreciation and thanks to my Supervisor Prof.Luigi Atzori for the opportunity that he gave me to participate in QoE-Net project. The research conducted during my doctorate was funded by QoE-Net MSCA-ITN project.

I want to express my sincere gratitude Prof. Luigi Atzori and my Co-Supervisor Prof.Virginia Pilloni for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of doctorate.

I want to thank my fellows from QoE-Net project for the stimulating discussions and for all the fun we have had in the last three years. Also, I would like to thank Prof. Periklis Chatzimisios from International Hellenic University, Mr Bernhard Feiten from Deutsche Telekom AG and Mr Martin Adolph from ITU for their support on my research during my internships in Thessaloniki, Berlin and Geneva, respectively.

Last but not least, I would like to thank my father Kostas, my mother Athina and my brother Christos and my friends, Giannis, Maria, Harry, Antonis, Kostas and Sofia for supporting me spiritually throughout my PhD life and my personal life.

Elisavet Grigoriou
Cagliari, 2019

Abstract

One of the major driving forces of the service and network's provider market is the user's perceived service quality and expectations, which are referred to as user's Quality of Experience (QoE). It is evident that QoE is particularly critical for network providers, who are challenged with the multimedia engineering problems (e.g. processing, compression) typical of traditional networks. They need to have the right QoE monitoring and management mechanisms to have a significant impact on their budget (e.g. by reducing the users‘ churn). Moreover, due to the rapid growth of mobile networks and multimedia services, it is crucial for Internet Service Providers (ISPs) to accurately monitor and manage the QoE for the delivered services and at the same time keep the computational resources and the power consumption at low levels.

The objective of this thesis is to investigate the issue of QoE monitoring and management for future networks. This research, developed during the PhD programme, aims to describe the State-of-the-Art and the concept of Virtual Probes (vProbes). Then, I proposed a QoE monitoring and management solution, two Agent-based solutions for QoE monitoring in LTE-Advanced networks, a QoE monitoring solution for multimedia services in 5G networks and an SDN-based approach for QoE management of multimedia services.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recent reports on the usage of data services [1] present a significant increment in data traffic by 60% from 2015 to 2016. This increment is due to the increment of the mobile subscriptions and the increased data traffic per subscriber, which is mostly triggered by the evolution of the applications and the Internet of Things (IoT). Moreover, cellular connectivity becomes a vital access methodology due to the increase of pervasiveness of mobile broadband. Until then, ETSI released an improved version of LTE which is considered as "4.5G", called "LTE-Advanced Pro" (3GPP Release 13) [2], and co-exists with the existing LTE networks. LTE-Advanced Pro (LTE-A Pro) has increased the data rates and the spectral efficiency compared to LTE. Moreover, it has become a significant trend on the networks domain, and a high QoE will come with excellent 5G experience by LTE-A Pro networks. In 5G networks, operators will employ efficient and cost-effective solutions in order to preserve CAPEX and impact users' QoE. Furthermore, 5G will be designed to utilize the scarce spectrum more efficiently. New management schemes and resource allocation techniques will be applied for this improvement. Moreover, the 5G networks focus on a more personalized communication experience using network adaptation and spectrum choice to user requirements and experience.

Future networks will rely on heterogeneous infrastructure that includes physical and virtual resources and a significant amount of multimedia traffic. The expected explosion of traffic will lead to inefficient utilization of resources which necessitates future network resources to be unified and dynamically pooled, as well as offered as-a-Service to multiple end-users. Furthermore, issues such as lack of flexibility and agility are going to be observed, especially in the case of new multimedia services such as telemedicine. In today's competitive environment, users have the option to choose from a variety of Service Providers because their expectations have increased. Therefore, service availability is not enough anymore. Now, they need to consider as key metrics, the end-user experience levels by measuring the efficiency and effectiveness of the service, and the service levels for availability, as contributing factors to the end-user experience.

Thus, Service Providers must deliver their services in such a way that users will thoroughly enjoy a rich experience at a logical price with an improved Quality of Experience (QoE).

The European COST Action IC1003 " QualiNet" defines QoE as [3]: *"Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfilment 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."* Thus, QoE is defined as the measure of how well a system or an application meets the user's expectations. QoE is different from Quality of Service (QoS), which focuses on measuring performance from a network perspective. With no doubt, QoE is directly related to QoS. In the case of multimedia engineering problems (e.g. processing, compression) in traditional networks, there is a challenge for Service Providers. They need to have the right QoE monitoring and management mechanisms to have a significant impact on their balance sheets (e.g. by reducing the users' churn). According to Qualinet [3], the QoE is influenced by many factors which can be categorized into three groups namely, Human (e.g., related to demographic characteristics), System (e.g., related to technical quality characteristics of an application/service) and Context (e.g., related to user's environment). Considering the bandwidth increase that is expected in future networks, such as 5G networks, and the rapid expansion of multimedia services, it is evident that, the end-users demand for even higher quality media content will also be increased.

Moreover, due to the rapid growth of mobile networks and multimedia services, it is crucial for Internet Service Providers (ISPs) to accurately monitor and manage the QoE for the delivered services and at the same time keep the computational resources and the power consumption at low levels. Moreover, during the last years, ISPs are trying to provide the best service that the network allows and to estimate perceived QoE of end-users. For QoE estimation and optimization, the operators monitor QoE at the application level. Also, they monitor the network performance to identify network-related QoE performance issues, retrieve application-aware and location-aware QoE data, complemented by network Key Performance Indicators (KPIs). Many factors influence the QoE of a user such as the time, the kind of application, user's device, user's location. So, there is a need for flexible monitoring and management systems that take into account these influence factors, without increasing the cost and avoiding customer churn. They should adopt a QoE monitoring solution that fulfils the following requirements [4]; (a) monitor parameters related to applications, network, user and context, (b) implement techniques to acquire user's preferences or expectations for a service and (c) make available the monitoring results to use them for the network and

QoE management.

The objective of this thesis is to propose QoE monitoring and management approaches for future networks such as the QoE-Agent and the virtual Probes.

1.2 Thesis structure

In Chapter 2, I describe the State-of-the-Art and the main concepts used in this thesis. At first, I describe the 4th Generation of networks and more precisely about the LTE-Advanced networks, their architecture and its components. Similarly, I describe the characteristics of 5G networks with emphasis on their flexibility that leads us to the softwarization concepts. With these in mind, I identify the advantages and disadvantages of virtualization technology. At the same time, I describe the SDN and the NVF concepts, Cloud computing and Mobile Edge Computing concepts, and I mentioned the advantages and disadvantages of them. I continue with the QoE definitions and the QoE assessment methods (e.g. objective and subjective). I mention the stages of a monitoring process and the methodologies (e.g. active and passive). Then, I describe the QoE monitoring process, with an emphasis on 5G networks, and I describe the methodologies that used (e.g. offline and online). QoE influence factors are described afterwards. Moreover, I describe the ARCU model and the QoE Agents that used during this thesis. In the end, I describe the procedure of troubleshooting, the concept of Big Data and Social IoT that are used in my QoE monitoring solutions.

In Chapter 3, I describe the State-of-the-Art regarding the concept of Virtual Probes (vProbes). The network virtualization allows the development of new monitoring tools such as the vProbes in the form of VNFs and they can be used to monitor the traffic between network elements. The vProbes deployed in any time and place and they have common features with the physical probes. However, they can be deployed in virtual and physical environments. In 3 based on the literature review, I present a taxonomy of the applications domains that a passive vProbe can be applied. I have identified four application domains: network monitoring, QoE monitoring, troubleshooting and other cases such as trace information leaks and malware behaviour. In each domain, the authors follow a "software-based" approach or a "hybrid" approach.

In Chapter 4, I describe the QoE estimations models that were used during this thesis. In most of the cases, the application was a video streaming service. Thus I used the same model or variants of it. First, I describe the "eMOS" model that is a QoE psychometric model. Then, I describe the Task, Network and Task Assignment model that I proposed and used in Chapter 6. Afterwards, I describe the QoE estimation model that was used by the QoE-Agents in Chapter 7. Then, I describe a model that extends the model, as mentioned earlier by considering accuracy. In the end, I describe a QoE estimation model which is a combination of Task Assignment model and the

QoE estimation model used by QoE Agents.

In Chapter 5, I describe the concept of vProbes. The network virtualization allows the development of new monitoring tools such as the virtual probes (vProbes) in the form of Virtual Network Functions (VNFs). The vProbes deployed in any time and place that the network provider needs them [5]. They have common features (are described in Section 5.2.1) with the physical probes, but they can be deployed in virtual and physical environments.

vProbes in the form of VNFs are used in many cases for the following reasons [6]:

1. **OPEX/CAPEX reduction**, through reduced equipment costs and reduced power consumption. The reduction of power consumption comes from the fact that VNFs can be turned off automatically when they are not needed. From the operator's perspective, vProbe removes the need for private probe appliances such as network taps.

2. **Scalability/Flexibility**. The operators can rapidly increase network analytics capacity when needed, by adding a vProbe instance on additional VMs on the same hardware, keeping CAPEX/OPEX under control. The vProbes are scalable and they can automatically "scale-up or scale-down" as needed with the other NFV infrastructure being monitored, offering network elasticity.

3. **Support of virtualization**. Many network operators are moving to virtualization. Thus, the problem with legacy physical probes is that they do not have access to internal VM-to-VM communication between network elements hosted on the same server because they are hardware-based and they do not have any knowledge on the virtualized network.

The main objectives of this research were to investigate the role of vProbes, to explore the literature, and observe the applications domains where a vProbe can be used. Based on this analysis, in this work, I provided a comprehensive definition of vProbe. I investigated the vProbes, their differences and similarities with the physical probes, their requirements, their functionalities, their challenges and I proposed an NFV-based architecture in order to describe how a vProbe can be used in a real scenario. Moreover, I used the concept of vProbes in the QoE monitoring and management solution in Chapter 6.

In Chapter 6, I describe a QoE monitoring and management solution. Given the fact that the estimation and modelling of QoE have gained considerable attention among service providers and network operators in the last years, properly designing a QoE monitoring and management infrastructure is necessary [7]. Indeed, providers are interested in QoE because the competition has grown and they need new aggregated-value solutions to decrease the number of clients who are churning because of quality dissatisfaction. Thus, a new flexible framework is proposed to address the need of network providers to improve user satisfaction and decrease churn. Therefore, in Chapter 6, I presented my solution that includes a framework, an SDN-based architecture and a platform. The proposed framework considers many functional requirements (in terms of QoE Influence Factors (IFs) and a QoE-related parameter (user satisfaction). Then I applied my proposed framework to an SDN-based architecture, and then the architecture is deployed in the proposed QoE Monitoring and Management Platform, called "QoE Mo-Ma".

The advantage of this solution is that it is flexible and can be used in a multi-application multi-context scenario. Using the proposed strategy, the operator can *plug'n'play* the QoE models that choose in order to estimate the user's satisfaction. When demonstrating the efficacy, I targeted to evaluate the QoE level of an adaptive video streaming service. In order to do that, I answered three questions: (i) item Which information needs to be collected?, (ii) From where I retrieve information?, and (iii) How will I do that?. The "Which information" is determined by the selected QoE parameter (in my case is the satisfaction). "Where" and "How" depended from the approach that I had implemented (e.g. SDN-based approach).

In Chapter 7, I describe two Agent-based solutions for QoE monitoring. I employ distributed monitoring agents, called "QoE Agents", in the form of Virtual Network Functions (VNFs) that monitor different QoE influence parameters.

The objective of the first solution [8] was to propose a strategy applied in an LTE-A Pro network, in order to estimate user's QoE in real-time using QoE-Agents based on a QoE-layered model. Our strategy considers the accuracy of the measurements, the network load, as well as the application and network parameters. The contribution of this work is that it presents the flexibility of the QoE-layered model, since it is not restricted to a specific estimation model, but it can also include many models depending on the needs of the network provider. Also, it shows if the QoE-Agents are an efficient solution for a network provider in order to monitor the QoE level. The novelty in the technical part of the proposed solution is the implementation of the evaluation system, which identifies the most suitable monitoring frequency.

In the second solution [9], the primary target was to estimate user's QoE by considering different Influence Factors (IFs) such as those related to the application, context and network and propose a monitoring algorithm. At the same time, this QoE monitoring solution satisfies the following two requirements: i) the desired level of accuracy in the monitoring results and ii) desired limit in the required resources for the monitoring. The proposed monitoring procedure measures network load, computational resources as well as network latency and then dynamically adjusts the monitoring frequency. The aim was to reach the desired level of accuracy of the QoE estimations avoiding an unnecessary increase in the usage of computational resources and network load caused by the monitoring procedures themselves. The proposed solution and algorithm applied to an LTE-Advanced Pro network since most of its components can be virtualized and extended based on ISP's needs. The advantage of the proposed approach is its capability to adjust the monitoring frequency while considering the accuracy of the QoE estimations.

In Chapter 8, I propose an SDN-approach for QoE management of multimedia services using resource allocation. I based on the concept of "network softwarization" that uses Software-Defined Networks and Network Function Virtualization (NFV) to execute any network functions and networked services as applications on Virtual Machines (VMs) which are allocated dynamically on general-purpose hardware. SDN and NFV are considered as key enabling technologies and an appealing solution for 5G networks. Motivated by the opportunities provided by these technologies, this work presented an SDN-based approach for QoE management of future multimedia services through QoE-driven network path assignment. I applied task-level scheduling intending to assign tasks related to a multimedia application to network elements. To achieve this goal, I developed a QoE-driven dynamic task allocation scheme for adaptive video streaming over SDN/NFV enabled networks. After all, I wanted to deploy new applications, services and infrastructures in order to meet future changing business goals while succeeding high level of end-users' QoE.

In Chapter 9, I propose a QoE monitoring solution for multimedia services in 5G networks that includes QoE agents [8] in the form of Virtual Network Functions (VNFs) in order to monitor user's satisfaction while decreasing the user traffic and the complexity during the QoE monitoring. The QoE agents are deployed in different network locations, and they can collect QoE related parameters and based on them to estimate the QoE level using the most suitable QoE models. The QoE agent's relocation from their conventional places to other places can provide economic benefits and can improve flexibility and scalability. The QoE agents can be deployed in the Network Edge using the Mobile Edge Computing (MEC) paradigm, in a Data Center using Cloud Computing paradigm, in the Core Network, in user's premises or they can be distributed. Thus, in this work, I proposed a QoE agent placement solution in order to monitor the QoE level for a variety of services. In this work, I proposed a model to optimize the QoE agent's performance. Their performance can be improved by the proposed algorithm, which efficiently selects the lowest number of monitoring locations that reduce the complexity and the traffic during a QoE monitoring. I formulate the problem of QoE agent placement problem in a "QoE-monitoring" scenario for an adaptive video streaming service. Preliminary results show that the most cost-effective and efficient approach is to follow a distributed approach and deploy the QoE agents in NFV-capable network devices and the Cloud. By following this approach, I can leverage from the real-time insights.

In Chapter 10, I describe our conclusions and mention our future work.

1.3 List of Publications

1. Grigoriou, E., Saoulidis, T., Atzori, L., Pilloni, V., & Chatzimisios, P. (2018, September). A QoE monitoring solution for LTE-Advanced Pro networks. In 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (pp. 1-5). IEEE.

2. Grigoriou, E., Saoulidis, T., Atzori, L., Pilloni, V., & Chatzimisios, P. (2018, May). An agent-based qoe monitoring strategy for lte networks. In 2018 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

3. Grigoriou, E., Barakabitze, A. A., Atzori, L., Sun, L., & Pilloni, V. (2017, May). An SDN-approach for QoE management of multimedia services using resource allocation. In 2017 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.

4. Grigoriou, E., Atzori, L., & Pilloni, V. (2017, December). A novel strategy for Quality of Experience monitoring and management. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-6). IEEE.

5. Grigoriou, E., Atzori, L., & Pilloni, V. . Virtual Probes: a Survey of the Application Domains, Technologies and Challenges (2019, Under Review)

6. Grigoriou, E., Atzori, L., & Pilloni, V. . A QoE monitoring solution for multimedia services in 5G networks (2019, Under Review)

Chapter 2

State-of-the-Art and Main Concepts

2.1 4G Networks

3GPP Long Term Evolution (LTE) standard is part of the 4th generation of networks and became the fastest and more consistent technology compared to competing technologies such as WiMAX. It was designed to provide broadband high-speed data service with very low latency, and it is capable of supporting innovative new services, providing a high quality of experience.

Last years 5G networks brought an adequate answer to the exponential increase of mobile Internet traffic. LTE-Advanced Pro improves the users' experience because both bandwidth and spectral efficiency have increased, whereas the latency is reduced [10]. An LTE-based architecture includes the Evolved-UTRAN (E-UTRAN) and Evolved Packet Core (EPC) network, as shown in Figure 2.1.1. LTE-A includes the Evolved-UTRAN (E-UTRAN) and Evolved Packet Core (EPC) network. The E-UTRAN is responsible for all radio-related functions, and it is composed by eNodeBs (eNBs) which are interconnected with each other and to the EPC as presented in Figure 2.1.1. The EPC includes the following logical nodes [2]: (1) Packet Data Network Gateway (PGW) performs IP address allocation for the users and QoE enforcement, (2) Serving Gateway (SGW) is a local mobility anchor for data bearers when the user is moving between eNBs, (3) Mobility Management Entity (MME) is responsible for bearer management and connection management, (4) Home Subscriber Server (HSS) is a repository of all subscriber and service-specific information and contains user subscription data, (5) Policy Control and Charging Rules Function (PCRF) is responsible for policy control decision-making and controls the flow-based charging functionalities, (6) Traffic Detection Function (TDF) provides information about the detected traffic at the user plane, detecting the application ID and its description, (7) Subscription Profile Repository (SPR) is a database storing information related to network usage policies of a user such as charging related information and (10) Application Function (AF) provides information about the session characteristics such as media type.

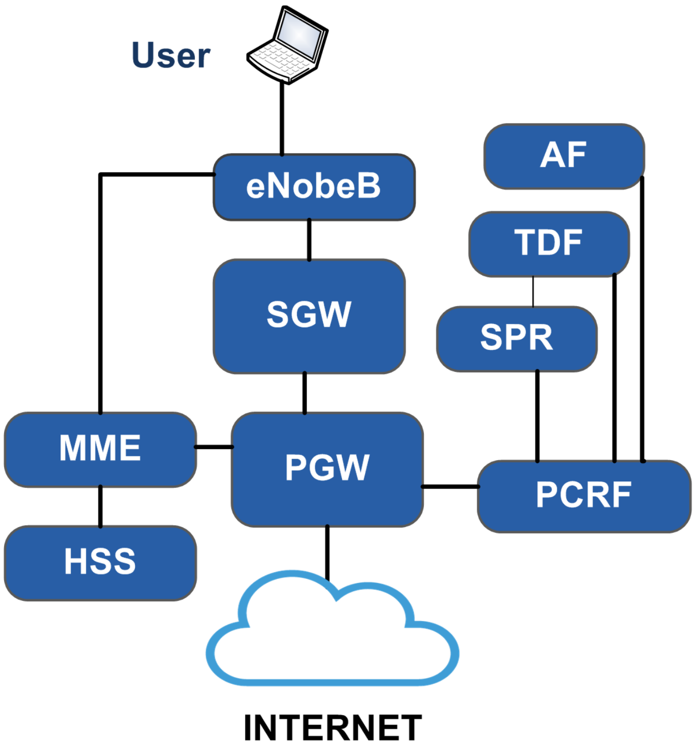


Figure 2.1.1: Overall LTE network architecture

2.2 5G Networks

5G is an evolving standard which was introduced by 3GPP and ITU. It is a successor to the current 4G cellular standards (LTE or LTE-A) to address the increase of traffic volume, the inflexibility of the hardware-based architectures at these networks. The vision of 5G services has been developed based on three scenarios [11], i.e. Enhanced Mobile Broadband, Ultra-reliable and low latency communications, and massive machine-type communications. The 5G specification includes technologies such as high-frequency mmWave base stations, beamforming and massive-input and multiple-output (MIMO), and changes related to infrastructure network slicing and data encoding. The evolution toward 5G mobile networks is characterized by an increasing number of wireless devices, increasing device and service complexity, and the requirement to access mobile services ubiquitously. Moreover, the 5G standard proposes better performance "Key Performance Indicators (KPIs)" [11], some of these are traffic volume density, experienced end-user throughput, latency, reliability, availability, retainability, energy consumption and cost.

Moreover, one of the main features of 5G is end-to-end flexibility. This flexibility will come from the concept of "network softwarization", the ability to create highly specialized network slices using advanced Software-Defined Networking (SDN), Network

Function Virtualization (NFV) and cloud computing capabilities. Virtual Network Infrastructures can be built faster and on-demand along with their VNFs. Meanwhile, there is a trend for decentralization of the cloud architecture, in the format of federated networked cloud (e.g. edge computing), which allows the placement of VNFs near to the end-users. By this decentralization, the end-to-end communication path can be reduced between the corresponding servers and the users, which can improve the user's QoE [12].

2.3 Softwarization concepts

The last years we are closer to the implementation of 5G networks that support virtualization of network functions and flexible placement, flexible deployment and management of resources, network programmability and multi-domain control management and orchestration and multi-tenancy support. The 5G networks are based on a comprehensive software system using all the available physical or virtual resources in a network. The network functions are becoming software only. Thus the network infrastructure is more flexible, and parallel deployments of multiple dedicated networks are enabled. The network functions can be installed in the Edge Network in computing nodes in order to provide differentiated services and resilience [13].

The ISPs can benefit from the NFV because they can increase their service value and their revenue streams. Also, they can have lower implementation costs and lower networking costs because the NFV can be placed in any NVF-able network device. Distributed Network Function Virtualization (DNFV) [14] approach for ISPs delivers new revenue opportunities via service agility, a quick introduction of new services, significant reduction of OPEX/CAPEX, better user experience and thus lower churn due to the fast and efficient delivery of services.

In Table 2.3.1, I have identify and present a few advantages and disadvantages of Virtualization technology.

Table 2.3.1: Virtualization technology

Advantages	**Disadvantages**
By virtualizing Core-Network Functions and Radio-Access network functions, I are able to reduce **deployment cost** and **operating cost** [15].	Not all the services and applications can be virtualized.
Operators deal with network traffic **dynamically** and in **real-time** [16].	It requires service chaining that must work efficiently.
VNFs are more **bandwidth-intensive** and **compute-intesive** [17].	It doesn't have the same performance as the physical resources.
More **flexible service deployment** due to easier service innovation [17].	VNF management is a challenging task.

Mobile networks evolved toward a software-defined infrastructure and apply virtualization concept at the network edge. This evolution leads us to reconsider the mobile network infrastructure and organization of the Radio Access Network (RAN) and the Core Network (CN) by introducing new use cases for network monetization in order to improve content delivery and user's experience [18].

The limitations of the traditional network technologies, such as lack of scalability, vendor dependency and network complexity, lead to the research of new technologies such as SDN and NFV. Nevertheless, the concept of changing the network's behaviour, to meet the application's requirements with efficient resource utilization, to improve user's satisfaction is not new [19][20] [21][4]. Thus, network providers need to deploy a service with the optimal set of network resources and, to do that, the concepts of SDN and NFV are introduced [22][23].

2.3.1 Software-Defined-Networks (SDN)

Moreover, the significant problems of current networking technologies are related to the limitations in scalability, the fact that is often vendor dependent and that do not support well dynamic service deployment [24]. The Software-Defined Networks (SDN) approach is considered as a possible solution to overcome most of these issues. Note that, one of the most popular deployed networks in the industry is Google's OpenFlow-based WAN, which is used as production network [25] presented at Open Network Summit 2012.

In traditional networks, the configuration of the control plane should be done in each network device, and this is time-consuming because multiple devices should be configured manually, and errors may occur. Nevertheless, traditional networks are static and do not support the dynamic, scalable computing and storage, which is needed by modern computing environments such as data centers [26]. These limitations lead to SDN that are the future network architectures to create **flexible programmable networks** through **software applications** that run on top of the SDN Controller and because the control plane decoupled from the data plane. The SDN Controller is a logically centralized entity that maintains a global state of the network. Thus, SDN supports the **centralization** and **abstraction** because they allow network administrators to divide and configure the network according to their needs without any knowledge of the underlying network.

An SDN is divided into three parts, (i) the southbound part provides the necessary interface between the SDN Controller and the SDN-based infrastructure (usually it uses the OpenFlow protocol [27][28]), (ii) the SDN Controller runs the Network Operating System (NOS) and includes the network and the control functions, (iii) the northbound part provides an interface between the SDN applications and the SDN Controller [29]

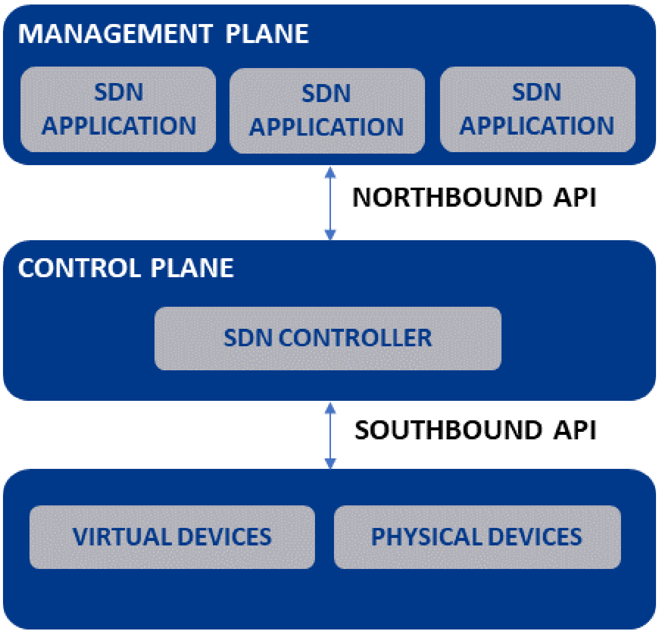


Figure 2.3.1: SDN-based architecture

[28]. Figure 2.3.1 describes the main components of SDN architecture.

2.3.2 Network Function Virtualization (NFV)

NFV aims at "Virtualizing Network Element or certain function of a network element in service provider network by using commodity machines" [30]. The motivation behind NFV is to use standard servers and commodity switches, to **reduce the costs to innovate services quickly, reduce power and space usage by using commodity servers and standardized and open interfaces**. The main parts of an NFV architecture are the "Network Function Virtualization Infrastructure (NFVI)", the "VNFs", the "Management and Orchestration" as shown in Figure 2.3.2. The NFVI [31] the software or hardware environment where the VNFs are deployed, and it includes the physical and virtual resources and the virtualization layer. The "Management and Orchestration" entity is responsible for creating of end-to-end services between VNFs and for resource management.

Both **SDN and NFV** focus on a software-centric network and try to leverage **virtualization and automation**. From one side, the SDN separates the data plane from the control plane, from the other hand, NFV decouples network functions from specific hardware elements. Moreover, SDN requires a new network construction, NFV can work on existing network resources, because it resides on servers and interacts with specific traffic sent to them. NFV does not need SDN for its implementation (and vice versa), but they can be implemented together [5][32][6].

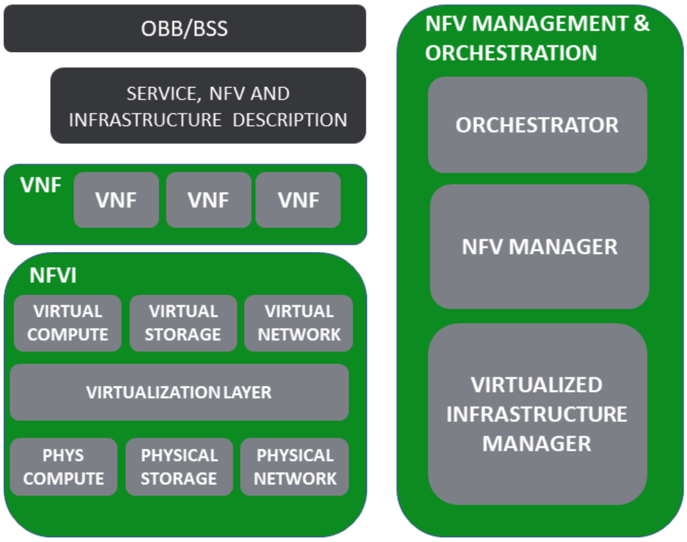


Figure 2.3.2: NFV Architecture

2.3.3 Mobile Edge Computing (MEC) and Cloud Computing (CC)

The SDN and VNF paradigms[18] help the network operators to manage softwarized network functions dynamically and transform their infrastructure using virtualization. Based on these concepts, ETSI Mobile Edge Computing (MEC) paradigm specifies a virtual platform at the mobile network edge that is called "MEC server" in order to host third-party players' applications. The MEC servers can be located at the Base Stations such as eNodeBs or aggregation points. The virtualized edge platforms are going to create new opportunities for network operators by leveraging the advantages of virtualization, including automation, flexibility. MEC paradigm will lead network operators to rethink the location of the network functions and will open the road for the collaboration between network operators and service providers.

In Table 2.3.2, I have identify and present a few advantages and disadvantages of MEC technology.

Edge Computing (EC) helps to ensure that the right processing takes place at the right time and place achieving lower latency and realizing real-time analytics. However, Data Centers achieve higher latency but are closer to business intelligence[38]. Furthermore, the Mobile Edge Computing (MEC) intends to improve the user's experience since it is close to end-users [39]. MEC is a layer between the Cloud servers and the mobile devices, as shown in Figure 2.3.3. Thus, the infrastructure is shown as a three-layer hierarchy [40]. The MEC servers collaborate with the Cloud servers to support and improve the performance of the end devices. As the bottom layer "Mobile devices" included the sensors, mobile devices, and social platforms which are in the Core Network through the Edge Network (i.e. MEC, RAN) and the Core Network is

Table 2.3.2: Mobile Edge Computing

Advantages	Disadvantages
Reduce end-to-end **delay** by moving the resources to the network edge [33] [34].	The **cost** of cloudlet provider depends on the amount and the location of the Cloudlets and their services [34].
Reduce **task execution time** and **energy consumption** by offloading tasks to centralized Data Centers (Cloudlets) [35].	It is resources constrained. It has **limited computational resources** because edge devices have smaller processors and limited power budget [36].
Support **Big Data analysis** in real-time [34]	It can be accessed by different radio access technologies. It needs to deal with the issue of network switch management.
More **flexible** because it allows the use of available computing and network resources and it has a impact on QoE and cost-efficiency [37].	It needs to deal with authentication, application and data protection and data integrity.
Less network hops thus it can have reduced **latency**.	

connected to the Cloud Network. Due to the LTE RAN evolution, it is more feasible to deploy MEC, which brings cloud services near to the mobile users. Thus, each edge platform represents an edge cloud with applications and services to a specific mobile environment.

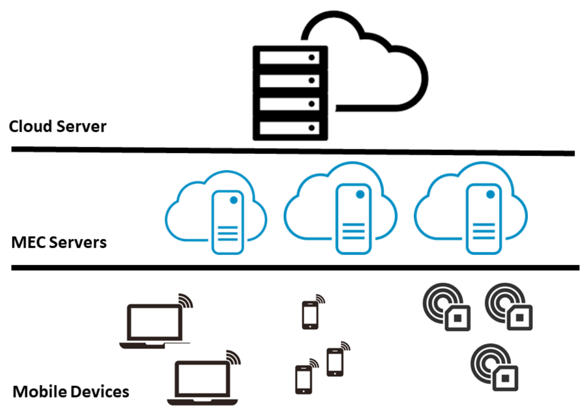


Figure 2.3.3: The three layer architecture of MEC

In Table 2.3.3, I have identify and present a few advantages and disadvantages of Cloud technology.

Based on the advantages and disadvantages of the MEC paradigm and Cloud Computing, I noticed that the ideal solution is a combination of these two for the following reasons:

- It supports location and context-aware services to improve QoE.
- It ensures scalability and load balancing to improve QoE.

Table 2.3.3: Cloud Computing

Advantages	**Disadvantages**
VNFs in Data Centers can help to facilitate the **dynamic service chaining** required to support the service and traffic demands of the network [17].	Large end-to-end delays [33].
VNFs in Data Centers can offer more dynamic **NFV deployment** and more efficient **utilization** of the network resources [17].	Increased latency and limited bandwidth prevent the use of real-time applications. It can hold the data but not react on real-time.
Store and manipulate huge amount of data.	High intercommunication latency between devices and Data Centers. This is solved by the Fog Paradigm [41].
Multi-level **authentication** to resolve security issues.	There is location awareness [42].

- It presents low latency and high capacity virtualized services and resources.

2.4 Quality of Experience (QoE)

There are several definitions for the concept of Quality of Experience (QoE); some of these are the following: **International Telecommunication Union (ITU)** defines QoE as “A degree of delight or annoyance of the users of an application or service ”[43]. Later, ITU [44] defined that the QoE is “The overall acceptability of an application or service, as perceived subjectively by the end-user”. This definition focuses on the acceptability of service and does not include the expectations or the context of the user.

ETSI defines QoE as “a measure of user performance based on both objective and subjective psychological measures of using ICT service or product”[45].

Qualinet [46] has introduced another definition about QoE which is “The QoE is the degree of delight or annoyance with respect to the utility and/or enjoyment of the application or service in the light of the user’s personality and current state [46].

The QoE assessment has several benefits that can help to prevent users’ churn, overcome service rejections, optimize services and can be technically expressed by linking to its QoS parameters[45].

ITU G.1011 recommendation provides estimation models that can be used to conduct QoE measurements on a variety of telecommunication services. The recommendation also mentions that the quality is not a unique QoE estimation factor that can be used to measure services. ITU divides the QoE assessment methods [47], i.e. objective assessment and subjective assessment. Subjective assessment is time-consuming, more expensive and requires special assessment facilities to produce reliable and reproducible results. The assessment is conducted using Mean Opinion Score (MOS). MOS is a scale of 1 to 5, which refers to the quality of telecommunication services according to user perception [48]. However, objective assessment predicts only from physical characteristics using estimation models which are used to measure QoE by involving QoS parameters. Both subjective and objective measurements are useful for network and service providers.

2.5 Monitoring

The monitoring process includes five **operations** [49] that are collection, representation, report, analysis and presentation. The "collection" process collects and formats raw data from the network, to forward them to the "presentation" process which includes the formatted data that can be used by different management functions such as to measure the network's behaviour, identify usage patterns in the network, verify the accuracy of configuration changes. The "report" process forwards the collected data from the network devices to other appliances, to be processed by other functions, the "analysis" process analyzes the data and extracts insights from them, and then, the "presentation" process presents the data to the network operator in different formats.

The monitoring process can be **active or passive**. In active monitoring, test traffic is injected into the network. It's possible to estimate network performance by tracking how the probe packets treated in the network. Nevertheless, it adds significant overhead to the network [50]. In the case of passive monitoring, the network performance derived from the supervision of the existing user traffic [51]. In comparison with active monitoring, passive monitoring does not add any overhead to the network and allows the processing of local traffic states and the behaviour of traffic flows passing through specific network locations. Nevertheless, it requires full access to network devices, and this can cause privacy and security issues [50]. There are many metrics to be monitored in a network, and they depend on the offered traffic, the network characteristics, the traffic characteristics, the network performance and the node performance.

2.5.1 QoE Monitoring

QoE in a 5G network is the level of user satisfaction that can meet the needs of the ITU usage scenarios (i.e. Enhanced Mobile Broadband, Ultra-reliable and low latency communications, and massive machine-type communications). Thus, it needs to be monitored using passive or active monitoring.

There are two measurement methodologies to monitor a communication network: offline and online. The offline methods are mostly used in network diagnostics, and the online methods are used for live network monitoring. Their main difference is the computational complexity (higher for offline methods) involved together with the resulting accuracy (lower for online methods) [52].

In order to monitor the QoE level, different monitoring techniques can be used, such as monitoring probes which are used to collect the data required in order to conduct a better QoE analysis. Monitoring probes can be divided into two types, passive and active. Passive probes inspect traffic that passes through them without interfering. Active probes capture a sample from an entire network that includes more detailed information [53].

The traditional (physical) monitoring techniques such as physical probes, present several limitations such as hardware dependence, lack of adaptation, an increment of OPEX and CAPEX and inability to monitor virtual environments. However, using virtualization technology, these limitations are solved using common device standards, the software is replacing the hardware, and it is programmable, and virtual network devices are used [54]. With these in mind, by applying virtualization technology, the ISP is more flexible because it is possible to instantiate services such as QoE monitoring, at any time. The shift to virtualization has to lead the ISPs to be more flexible, hardware-independent and reduce OPEX/CAPEX. Moreover, distributed NFV (DNFV) [14] approach for ISPs delivers new revenue opportunities via service agility, a quick introduction of new services, significant reduction of OPEX/CAPEX, better user experience and thus lower churn due to the fast and efficient delivery of services.

2.5.2 QoE Influence Factors

Regarding the QoE, in [55] proposed the Application-Resource-Context-User (ARCU) model. In ARCU model, the authors identify and categorize QoE Influence Factors (IFs) in 4 multi-dimensional spaces as shown in Figure 2.5.1, which are: (i) Resource Space includes dimensions related to technical system properties and network resources used for the service, (ii) Application Space includes dimensions related to service/application configuration factors, such as resolution, (iii) Context Space includes dimensions related to the situation that the service/ application are used such as time of usage, and (iv) User Space includes dimensions related to specific user. The factors considered in this space are related to demographic data, expectations. However, the ARCU Model does not provide any hierarchy between these spaces. Thus, [56] proposed a QoE-Layered model that includes six layers; Resource, Application, Interface, Context, Human and User. Each layer of the ARCU model mapped to one or two layers of the QoE-Layered model, as presented in Figure 2.5.1 [56]. Each layer in QoE-Layered model includes a vector and a process. The input of the process is the vector with the internal parameters which are related to each layer and a vector with external parameters. The process transforms these parameters into a parameter vector. The process considered as the objective quality function with external and internal parameters. The output vector of a layer becomes the input vector of the next layer in the QoE-layered model. A quality value can be associated with each layer. Thus, in the QoE-layered model, six layers are defined in terms of processes, parameters, input and output.

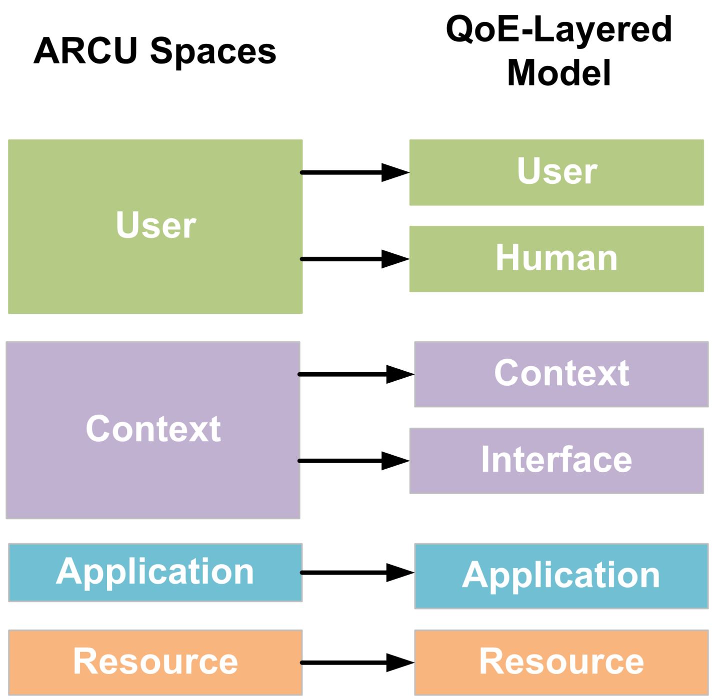


Figure 2.5.1: Mapping the ARCU spaces onto the QoE-layered model

2.5.3 QoE Agents

The authors in [56] have implemented a QoE-Agent based on QoE-Layered model, which can be integrated into legacy management systems. The objective of this agent is to obtain useful information about the QoE of any service implementing the model. Moreover, QoE-Agent does not specify any quality model but allows the user to plug their own models in any of the six layers. This agent uses passive probes to retrieve the necessary internal parameters at each layer using new or existing tools dedicated to the agent. It's important to mention that there are two approaches for QoE-Agents, the "Stand-alone Agent" that implements all the components and the "Distributed Agent" that includes a "Master QoE-Agent" and a "Slave QoE-Agent", in which the components are implemented by the Master or Slave agents. Any type of QoE-Agent shall implement the following components: Communication, Data-acquisition, Controller and Timer. If the QoE-agent is "Master-Agent" shall also implement the "Persistent-data" component otherwise it's called "Slave QoE-Agent".

2.6 Troubleshooting

Troubleshooting is a procedure with the target to **resolve problems and restore the network operations** in a network. Collective measurements and processes used to identify, diagnose and resolve problems and issues in a network. Currently, there are many different applications, network protocols and administrative domains; thus, the troubleshooting performance became more complex. Thus, network probes can be installed in network edges, at the ISP's side or the user's side. In real networks, open-source software such as NetProbes [57] can be used for troubleshooting or in case of SDN-based network-specific tools can be used such as SDN-RADAR [58].

2.7 Big data

The usage of "Big Data" can help the network operators to prevent network problems and improve the perceived user experience by collecting and analysing the vast and diverse amount of data [59]. The main characteristics of big data are velocity, variety, volume, veracity and value. ITU claims that mobile network traffic data have these characteristics. Thus these data can be used to estimate, model and monitor QoE in heterogeneous environments.

2.8 Social Internet-of-Things (SIoT)

Considering the user's context and how it's possible to retrieve information about it, I have to mention another concept which is the "Social Internet-of-Things (SIoT)"[60]. The SIoT is a paradigm according to which connected objects to the Internet create a dynamic social network that is used to route information and service requests, disseminate data and evaluate the trust level of each member in the network. The SIoT is based on the interoperability between systems and introduces the idea of social relationships between different devices without considering if they belong to the same platform or not and if different organizations [60] manages them. Thus, I benefit from the social relationships between devices in order to identify a user's context.

Chapter 3

State-of-the-Art on virtual Probes

The network virtualization allows the development of new monitoring tools such as vProbes. For this reason, I conducted a literature review, and I allocated this chapter to describe the taxonomy of the applications domains that a passive vProbe can be applied to, which are described in Figure 3.1.1. Based on the literature review, I have identified **four application domains**: network monitoring, QoE monitoring, troubleshooting and other cases such as trace information leaks and malware behaviour. In each domain, the authors follow a **"software-based" approach** ("orange bubbles" in Figure 3.1.1) or a **"hybrid" approach** ("blue bubbles" in Figure 3.1.1). In the case of software-based approaches, the authors do not use any traditional (physical) hardware equipment, but they use only vProbes. On the contrary, in hybrid approaches, the authors use vProbes and additional hardware monitoring devices. The main advantage of the hybrid approach is that it can be used in physical and virtual networks, allowing the network providers to be more flexible, using their existing (physical) infrastructures. Even though the hybrid approaches use both physical and vProbes, in my current work, I only focus on vProbes, to narrow the scope of the work. Note that all the surveyed approaches, except [61], are following a passive-based monitoring process. Indeed, there are not many approaches in the literature related to active monitoring using vProbes.

3.1 Application Domains

In Figure 3.1.1 for each related work from the literature, I have identified the probe type, the used technologies and the network type. The type of probe could be software-based or hybrid-based. The used technologies are SDN, NFV, virtualization such as using VMs. The network type could be either a physical or a virtual network. The network type and the probe type are related, meaning that, in case a network provider has a physical and virtual network infrastructure, the vProbe type used is "hybrid". Note that, the references with "*" are commercial products.

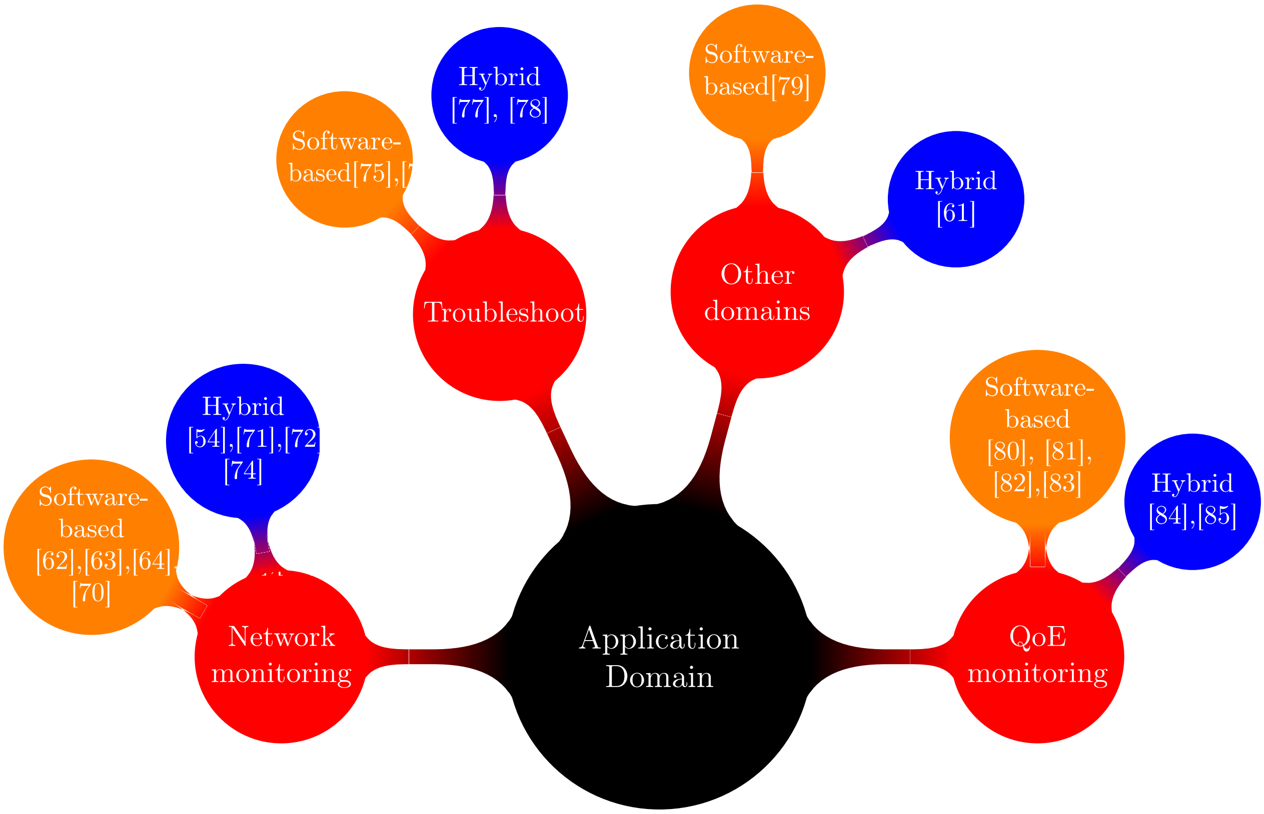


Figure 3.1.1: The Application Domains and the proposed approaches based on the literature.

3.1.1 Network Monitoring

Over the last years, many methodologies and tools have developed for network monitoring for physical networks, to estimate the performance of the network and user's behaviour [88]. The adoption of virtualization technologies by the network operators has led the networking community to develop new tools and methodologies to monitor virtual environments. A vProbe can measure [89]:

- network layer parameters such as jitter, packet loss, delay and packet size
- physical layer parameters such as throughput, bandwidth and bit rate
- additional parameters such as session establishment delay

I should note that there are similar functionalities between physical probes and vProbes, but the difference is the deployment environment of the probes. The functionalities are described in details in Section 5.2.1.

With software-based or hybrid approaches, network providers can monitor their network using a probe that is inside a virtualized environment, leveraging all the advantages of virtualization technology. On the contrary to software-based approaches, hybrid approaches also include physical probes that use traditional technologies such as sFlow [90] NetFlow [91] and RMON [92], to monitor the network. Nevertheless, the monitoring solutions that use vProbes are preferred from the network providers because they can be quickly deployed based on their needs. Moreover, vProbes are more flexible in comparison with physical probes, because in case a network provider needs to monitor a specific area of its network, it can instantly run a new vProbe in any location (e.g. a new VNF) without the need for a dedicated hardware device. Also, the use of a vProbe has lower costs for network providers than the use of a dedicated device.

Flexible and adaptable architectures can address the current and future business needs, deal with the newly offered services and support network adaptability considering the network condition. These new flexible and adaptable architectures should be introduced to support rapid service deployment and migration in heterogeneous network environments without considering the infrastructure type, using advanced monitoring capabilities and deploying vProbes in an ad-hoc manner [66]. Moreover, a new architecture component can be used to deal with QoE-awareness, which is facilitated using virtualization [64] or a component to build scalable, secure and reliable SDN architectures providing monitoring and management abstraction [67].

Software-based approaches

A software-based approach preferred from a hybrid approach in case a network provider aims to monitor only its virtual network. For network monitoring in an SDN environment, the vProbes are configured with specific characteristics, to be synchronized, monitor more interfaces, redirect user's traffic and collect and aggregate statistics [49]. In an SDN environment, the vProbe can use the SDN data plane to control the network using information from virtual switches (e.g. Open Virtual Switch) [63]. Nevertheless, in an NFV/SDN environment, a vProbe can be deployed inside a vCPE [68] [76]. It can add and process monitoring metadata, offering the benefits of flexibility, scalability and elasticity to the network provider, due to the virtualization of the monitoring resources [76]. In contrast to hybrid approaches, software-based approaches are preferable when there is no need for a dedicated Virtual Machine (VM) for the deployment of a vProbe that confirmed by software-based approaches such as [68], which are available on the market.

Hybrid approaches

In hybrid approaches, the operator can monitor its virtual and physical network. Nevertheless, the advantage of the hybrid approaches is that the operator can monitor its physical network by applying physical probes. Thus it needs to deploy the vProbes in its virtual network. In contrast with software-based approaches, in hybrid approaches proposed by the market, a dedicated VM is used by [71] and [72] commercial products.

Since in his two cases, the virtual probe is self-contained, consumes the platform's resources and requires some organizational actions. Note that sit's not mandatory to use only dedicated VMs in hybrid approaches, but the [71] and [72] follow this approach. A vProbe is a virtual appliance in a VM and includes an Operating System (OS) and a virtual software agent [71]. Thus, this VM (with the vProbe functionality) can run instantly in a physical or virtual environment. Additionally, the network operators can have extended visibility into their applications and services, contained in a virtual computing environment. However, a vProbe can also be deployed as VNF in a virtual environment, can collect data and apply simple analytics [72], and it's more flexible to monitor the network performance using traffic protocol inspection and record Packet Capture (PCAP) files, which include the captured network traffic collected by a packet sniffing API [73]. In hybrid approaches, VMs must include the monitoring agent and operating system, to be able to monitor the physical and virtual network infrastructure effectively. The vProbe is adjusted based on the network type. Moreover, a Software Defined Monitoring Controller (SDMC) as VNF in 5G core network that deploys virtual probes in every VNF can be used in addition to the physical probes in physical hardware components [54]. Therefore, network operators can reduce their hardware costs and the complexity of their network. Moreover, they can resolve network performance issues faster, can gain real-time insights to improve the QoE level of end-user and reduce customer's churn.

3.1.2 QoE monitoring

In the QoE monitoring application domain, the collected data can be made of network-related parameters, equipment factors such as codec type, or service-dependent parameters such as video resolution and web-page download time [89]. These data can be used to extract useful insights related to customers' experience. Nevertheless, the QoE monitoring domain incorporates network monitoring because network conditions have an impact on the QoE level. The QoE metrics can be subjective such as Mean Opinion Score (MOS) or objective such as video resolution. QoE metrics are used to monitor and estimate the user's satisfaction for a particular service. QoE metrics such as delay, or charging policies and costs can be related to a specific application such as video application or Video-on-Demand [89].

It's important for QoE monitoring to identify the network traffic type using traffic identification techniques such as Deep Packet Inspection (DPI), support QoE (objective or subjective) metrics for standardized protocols and support of Over-The-Top (OTT) services such as Netflix and Amazon Prime [84]. The network traffic identification helps the provider to choose the proper QoE estimation model in order to evaluate the user's experience. In hybrid approaches, external physical or virtual appliances can be used, such as DPI, in collaboration with the vProbe in the virtual environment to identify the type of traffic and accomplish a better QoE estimation. Moreover, the vProbes are preferred to physical probes because they can monitor and then estimate the QoE level of end-users [80],[81] because they are more flexible and can easily be deployed in different locations in a network. Consequently, hybrid approaches are more suited for QoE monitoring, because they can retrieve information from physical appliances using, e.g. agents in the user's equipment, to accomplish better estimations. But the opposite is not feasible: physical probes are not able to monitor and retrieve information from virtual environments. Note that, in both software-based and hybrid approaches, the vProbe can be implemented in the edge network in order to achieve higher accuracy [81].

Software-based approaches

A vProbe can be deployed independently as a VM running in an NFV environment, and it can receive copies of the monitored traffic, to estimate the QoE level. However, it is essential to support automatic deployment, centralized data collection, and incorporate a correlation system, to extract useful insights. The data are collected through an API (such as REST API) by the SDN Controller, and then KPIs are extracted per call/stream QoE metrics [82]. The vProbe can derive application statistics such as video start-time, to perform customer-experience analytics on a network service [80]. It can be used to measure the quality of an application (e.g. video application) and estimate the QoE at the user's side. Some examples provided by the commercial approaches [82] and [83].

Hybrid approaches

The network operators must monitor their virtual or hybrid networks, to be more flexible. A probe, as the probe in [71], is a self-contained virtual appliance and can be installed easily in an environment, as a VM, with the ability to monitor virtual or hybrid networks. In this case, the existence of an NFV infrastructure is not mandatory. The network operator can apply this hybrid probe using a VM in any location it's needed in its network. Each virtual appliance in the virtual environment includes an OS and vProbe functionalities. Nevertheless, the aim of both types of probes (physical or virtual) is to monitor QoE-level and use real-time metadata extraction for QoE estimation. Furthermore, network-related or essential application-related performance parameters can be collected by a distributed or centrally controlled system of vProbes, to estimate the QoE level [61] in a physical or virtual environment. Some examples are provided by the commercial approaches are [71][84] [85].

3.1.3 Troubleshooting

A vProbe can be used as a troubleshooting tool to monitor and analyze performance issues in applications and networks in a virtual or physical environment collecting specific logs. They are activated only when an unpredictable problem occurs [75]. In troubleshooting cases, the vProbe can detect the problems in the network measuring packet drops, delays.

Software-based approaches

In case of troubleshooting, a vProbe that follows a software-based approach can be used in an SDN/NFV environment to detect issues in the SDN data plane, e.g. issues related to virtual devices like virtual switches [76]. The virtual switches are distributed in the SDN data plane, and the vProbe can be used to detect unpredictable problems in it. In this case, the benefit of using a vProbe is that it is programmed dynamically and it can run when and where an issue is detected in order to resolve it.

Hybrid approaches

A vProbe can be used in virtual or physical environments for troubleshooting and service assurance cases. It is deployed as VNF for test, measurement or monitoring purposes by using active monitoring [61] to measure real-world services (in the physical infrastructure) using on-demand traffic generation [77] or using passive monitoring (in the physical and virtual infrastructure). Its objective is to gain IP network insights by the capture and the analysis of the network traffic in order to identify performance issues [77].

3.1.4 Other cases

The vProbes can be used, also, in other application domains such as to trace information leaks and malware behaviour using an Intrusion Detection System (IDS) [79] or for service chain monitoring [78].

Software-based approaches

In the cases mentioned above, vProbes can be used to enhance a system to support fine-grained and large-scale monitoring, to trace information leaks and malware behaviour in a virtual environment. The vProbe is placed in a separate VM using Virtual Machine Monitors (VMMs) and uses virtualization capabilities to discover information leaks. Besides, a vProbe can identify virtual network topologies, and traffic flows and this information is kept in the process memory of each vProbe. For example, this can result in an improvement of the accuracy of an IDS [79] in a virtual environment.

Hybrid approaches

A hybrid vProbe can be used for network tuning and optimization, or for early network problem detection. The hybrid approach is the optimal option because it enables the identification of possible trace information leaks or security attacks both in physical and virtual networks.

3.1.5 Physical or virtual probes? Software-based or hybrid-approach?

Based on the considerations presented above, *vProbes are preferred* in comparison with physical probes, because they are more flexible, adaptable and cheaper. Instead of physical probes, vProbes still have to deal with the accuracy of their measurements. In case of physical probe or vProbe, the time measurement accuracy and the clock accuracy must be considered, to make more accurate estimations in a virtual or physical environment. To sum up, the vProbes that used in the literature were presented and labelled as "software-based" or "hybrid". As can be seen for the previous subsections, in all the application domains **hybrid approaches currently are the optimal solution** to deploy a vProbe, because most of the network providers are under a "virtualization technology" adoption phase. Indeed, it's not feasible to replace all their physical infrastructures immediately with entirely virtual ones. Thus hybrid approaches give them the opportunity to support both infrastructures. Furthermore, hybrid approaches have the advantage that physical probes can be used together with vProbes: the network traffic gathered by physical probes forwarded to vProbes in the virtualized environment for further analysis.

Table 3.1.1: Related Works categorized by Application Domain (*Commercial products)

Application Domain	**Probe**	**Technologies**				**Network**	**Ref.**
		SDN	**NFV**	**virtualization**	**Physical**	**Virtual**	
Network monitoring	Software	X	X		X		[64]
Network monitoring	Software	X	X			X	[65]
Network monitoring	Software	X	X		X		[66]
Network monitoring	Software	X	X			X	[67]
Network monitoring	Software		X	X	X	X	[62]
Network monitoring	Software	X				X	[63]
Network monitoring	Software		X	X		X	[86]*
Network monitoring	Software		X	X		X	[70]*
Network monitoring	Software	X	X			X	[68]*
Network monitoring	Software		X	X		X	[69]*
Network monitoring	Hybrid	X	X	X	X	X	[54]
Network monitoring	Hybrid			X	X	X	[71]*
Network monitoring	Hybrid	X	X	X	X	X	[72]*
Network monitoring	Hybrid		X	X	X	X	[73]*
Network monitoring	Hybrid		X	X	X	X	[74]*
Network monitoring	Hybrid		X	X	X	X	[87]*
Network monitoring	Software	X		X		X	[76]
QoE monitoring	Software		X	X	X	X	[80]
QoE monitoring	Software		X	X	X	X	[81]
QoE monitoring	Software	X	X		-	X	[82]*
QoE monitoring	Software		X	X		X	[83]*
QoE monitoring	Hybrid			X	X	X	[84]*
QoE monitoring	Hybrid		X	X	X	X	[85]*
Troubleshooting	Software	X	X			X	[75]
Troubleshooting	Software	X		X		X	[76]
Troubleshooting	Hybrid	X	X		X	X	[77]*
Troubleshooting	Hybrid	X	X		X	X	[78]
Other cases	Software			X	X		[79]
Other cases	Hybrid			X	X	X	[61]

Chapter 4

QoE models

4.1 eMOS model

The monitored KPIs (e.g. video quality level) can be used to analyze the QoE of video sessions, and they provide information about the quality perceived and mapped to QoE using eMOS, which is a QoE psychometric model [93]. Video quality is chosen because it is a main influence parameter of the video QoE. In order to obtain this parameter, I monitor the resolution of the video, the rebuffering frequency, the duration of the rebuffering and the quality switch rates. "QoE-MoMa" platform evaluates the performance of the proposed QoE monitoring and management strategy applying all the components of it. Applying the eMOS model [93], I consider how important each parameter is with respect to the other and apply an appropriate coefficient to it. The rebuffering frequency and the representation quality switch rate measure the occurrences per second.

$$eMOS = \sum_{i=0}^{N-1} a_i \times x_i \tag{4.1}$$

Where

1. $\{x_0 ... x_{N-1}\}$ are the measured values of the mentioned metrics.

2. $\{a_0 ... a_{N-1}\}$ are the coefficients used for the eMOS calculation.

4.2 QoE model for resource allocation

4.2.1 Task Model

My task model considers, as an example, a video streaming service which includes four tasks: caching (source video), encoding, forwarding (it refers to one forwarding action running on one node) and playing back (client-side). A service can be decomposed into a set of tasks 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. Figure 4.2.1 illustrates an example of a general serial DAG where each node represents a task.

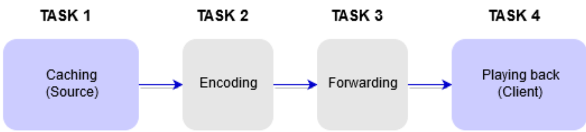


Figure 4.2.1: Example of a general serial DAG.

A binary vector $X_i = [x_{i\lambda}]$ for $\lambda \in T$, can be assigned to each node i in the network. $x_{i\lambda}$ is a boolean value representing the current state of the node i corresponding to task λ. When node i performs a task λ then $x_{i\lambda} = 1$. Figure 4.2.2 illustrates the DAG of a media streaming service task chain. Thus, each task should be executed in order to deliver the video from the media server to the end-user. Depending on the network and application parameters, my goal is to improve/optimize the overall QoE by making the best task assignment to network nodes.

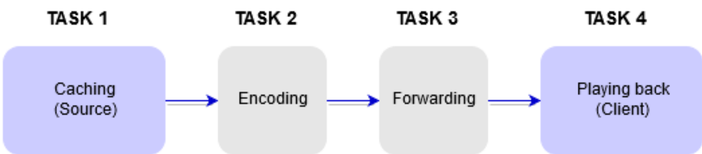


Figure 4.2.2: DAG for adaptive media streaming

4.2.2 Network Model

The network is modelled 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). Figure 4.2.3 illustrates the network model which depicts the SDN/NFV overview with the following components:

1. *Hardware Resources* which consist of Compute, Storage and Network modules. These are the physical resources related to CPU, memory and network, respectively.

2. *Virtualization Layer* abstracts the hardware resources and anchors the VNFs (Virtual Network Functions) to the virtualized infrastructure.

3. *Virtual Resources* consist of the vCompute, vStorage and vNetwork modules. The Data Plane includes the VNFs which are controlled by the SDN-Controller via Southbound API using the OpenFlow protocol. The SDN-Controller communicates with a "QoE management application "in the Management Plane via a

Northbound API, in order to monitor and manage the media flows in the Data Plane.

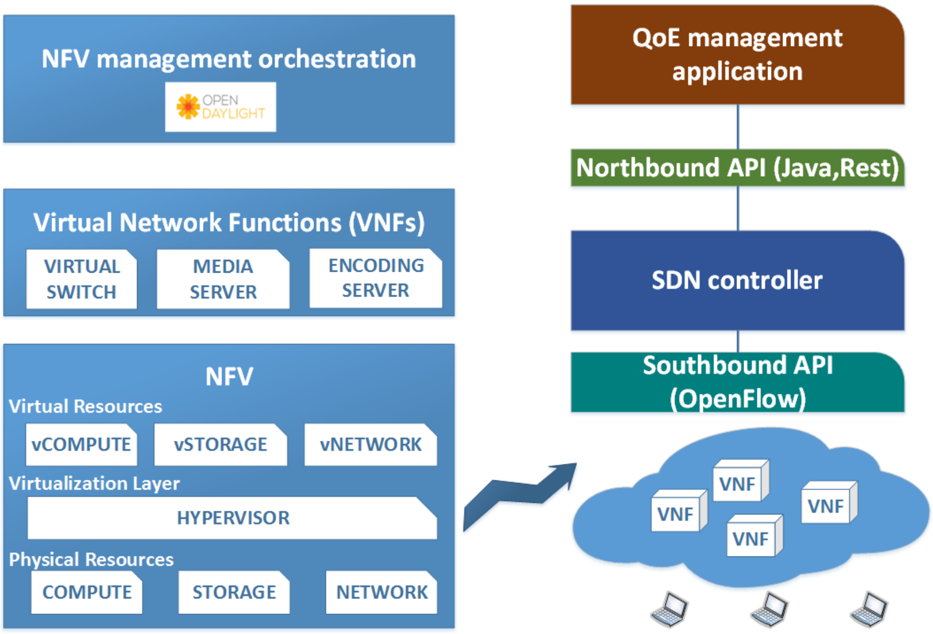


Figure 4.2.3: Network Model NFV/SDN view

Different services have different requirements and different parameters. Depending on user‘s requirements, each service can be divided into smaller tasks that can be assigned to different NEs (namely different VNFs) in order to deliver the video to the end-user using an Openflow-based virtual switch from a media server where the original version of the video is stored.

An optimization problem aims to find the best path of nodes to improve the QoE level. Although this is an NP-hard problem, which could be time and energy-consuming [94], I apply a centralized optimization approach trying to make 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 network is to maximize its own utility. My solution chooses a task assignment strategy x_i that maximizes utility function. Therefore, a strategy x_i^* is preferred to a strategy x_i, if and only if $u_{net}(x_i^*) > u_{net}(x_i)$.

4.2.3 Overall Utility Function

In order to formalize the correlation between network performance and user-perceived quality, a utility function is defined. The concept of the utility function is adopted from economics. It provides the means for reflecting a normalized and transparent way of various services performance prerequisites, users degree of satisfaction, different types of networks diverse resources and different QoS provisioning mechanisms and capabilities under common utility-based optimization problems [95]. My 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:

$$u_{net} = max \sum_{i \in Z} \sum_{\lambda \in T} (\alpha \times b_{i\lambda} - \beta \times c_{i\lambda}) \times x_{i\lambda} \tag{4.2}$$

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 [96]:

$$c_{i\lambda} = \gamma_i \times CPU_{i\lambda} + \delta_i \times Memory_{i\lambda}, \forall i \in Z, \forall \lambda \in T \tag{4.3}$$

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 [97] 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:

$$b_{i\lambda} = Q_r \times (1 - QoS(C))^{\frac{QoS(C) \times A}{R}} \tag{4.4}$$

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 with the premium service subscriber's requests is higher than normal service 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 [97], 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. 4.17. The $QoS(C)$ value can be calculated with the total sum of the values multiplying the measured QoS parameters in the network layer with the allocated weights. These weights are selected according to the type of the access network for the service. The considered QoS parameters are Packet Loss (PL), Packet Jitter (PJ), Packet Delay (PD) and Bandwidth (BW). The normalized QoS value reflects the network condition and is calculated as follows [97]:

$$QoS(C) = PL \times W_{PL} + PJ \times W_{PJ} + PD \times W_{PD} + B \times W_{BW} \tag{4.5}$$

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 is given from [97] 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

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

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

$$\sum_{\lambda \in T} x_{i\lambda} = 1 \forall i \in Z \tag{4.7}$$

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 $e_{ij} = 1$

$$e_{ij} = \begin{cases} 1 & \text{if } x_{i,\lambda} = 1 \text{ AND } x_{j,(\lambda+1)} = 1 \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in Z \tag{4.8}$$

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 $i \in Z$, I define a set of available resources as $Available_i = \{CPU_i, Memory_i, \dots\}$
For a task $\lambda \in T$, I define a set of required resources as $Required_\lambda = \{CPU_\lambda, Memory_\lambda, \dots\}$

$$Required_\lambda \leq Available_i, \forall \lambda \in T, \forall i \in Z \tag{4.9}$$

4.3 QoE estimation model used by QoE-Agents

Without loss of generality and in order to better present how the proposed strategy works, herein I consider a video streaming application. Thus, I then have considered some simple models for this application. However, the proposed strategy is model-agnostic.

Estimate network performance In order to estimate the network performance, I considered packet loss, packet jitter, packet delay and bandwidth. The normalized QoS value (QoS) refers to the network performance, and I propose to compute an overall QoS level as a weighted average of the QoS level for the major performance indicators as presented in Eq. 4.18. The QoS value calculated with the total sum of the values multiplying the measured QoS parameters in the network layer with the allocated weights. These weights are selected according to the type of the access network for the service. The considered QoS parameters are Packet Loss (PL), Packet Jitter (PJ), Packet Delay (PD) and Bandwidth (BW). The normalized QoS value reflects the network condition and is calculated as follows [97]:

$$QoS = PL \times W_{PL} + PJ \times W_{PJ} + PD \times W_{PD} + B \times W_{BW} \tag{4.10}$$

The above equation calculates the QoS for each path between users and the Application Provider. Where W_{PL},W_{PJ},W_{PD} and W_{BW} are the weights for packet loss, packet jitter, packet delay and bandwidth, respectively. These weights are assigned according to the quality standard bounds, and their relative importance degree is given from [97] as follows, PL 58,9%, PJ 15.1%, PD 14,9% and BW 11,1%. Their measured units were a percentage for PL, milliseconds for PJ and PD and Mbps for BW.

Estimate QoE A correlation model from [97] is used to map the QoS parameters to a QoE metric considering application parameters such as the video resolution, the Group of Picture (GoP) and the estimated video quality. The model is derived from a normalized QoS value as follows:

$$Est.QoE = Q_r \times (1 - QoS)^{\frac{QoS \times A}{R}} \tag{4.11}$$

Where A A is a constant related to the video resolution such as 240p, 360p, 480p or 720p, R is a constant which reflects the structure of the video frames according to the GoP length, and it is defined as $R = 24$ and Q_r is a constant factor that determines the overall QoE of video streaming service. Based on [97], the constant Q_r is set to 0.90.
To estimate the accuracy of the measurements, I used Mean Squared Error (MSE) metric which is the deviation of my estimated value from the true one and it is equal to the variance plus the squared bias. Moreover, if the sampling method and estimating procedure of the mean square error lead to an unbiased estimator, then the mean square error is simply the variance of the estimator. In order to estimate the computational complexity in the agent-based monitoring system, I measured the CPU utilization and the memory utilization of the devices that I located the agents each time that I conduct a measurement.

4.4 QoE estimation model used by QoE agents for the extended results considering accuracy

To estimate the QoE of the video service, the "eMOS model" [98] was used because it considers the network parameters (e.g., delay, bandwidth, jitter, loss) and application parameters (e.g. resolution, frame rate and video quality). Based on the analysis of video sessions, I can identify the quality perceived by the user and mapped it to QoE using eMOS, which is a QoE psychometric model. Video quality is chosen as a metric because it is a main influence parameter of the video QoE. The used video was the "Elephants Dream"[99] with a 480p resolution by using the H.264 codec, the frame rate at 24fps and mp4 as a container.

In order to prove the reliability of the algorithm, I calculate the accuracy of the QoE estimations with the Mean Squared Error (MSE) metric. I assume that the real QoE values are the QoE values measured every 1 second. In the static approach (without the usage of the proposed algorithm) the "Time between consecutive samples" was every 3 minutes and in the dynamic approach (using the proposed monitoring algorithm) the "Time between consecutive samples" is adjusted based on the network load and the computational complexity. The results with the accuracy using MSE metric when the algorithm is used (Equation 4.12) or not (Equation 4.13) are shown in Table 7.4.1. N is the number of samples in each measuring period. The total measuring period was 30 minutes. I observe that using the proposed solution, the MSE is lower. The aim was to avoid network overload and to consume fewer resources. Thus, the samples frequency is increased or decreased properly.

$$MSEwith = \frac{\sum (QoE_i^{1sec} - QoE_i^{AdjustedFreq})^2}{N} \tag{4.12}$$

$$MSEwithout = \frac{\sum (QoE_i^{1sec} - QoE_i^{3minlast})^2}{N} \tag{4.13}$$

4.5 QoE estimation model for the placement algorithm

4.5.1 Task Model

In the current work, a "QoE monitoring" procedure considered as a service. A service can be decomposed into a set of tasks and denoted as: $G_T = (T, E_T)$, $T = \{1, ..., \lambda, ..., \Lambda\}$ represents the set of tasks and $E_T = (e_{ij})$ is the set of edges such that each edge e_{ij} represents a unidirectional data transfer from task i to task j.

My task model, as shown in Figure 4.5.1, considers a "QoE-monitoring" service which includes QoE agents; the "Master QoE agent" (MQA) and the "Slave QoE agent" (SQA).

An MQA is able to collaborate with many SQAs, but an SQA is able to collaborate only with one MQA. Many SQAs can exist in network infrastructure and collaborate with only one MQA [56].

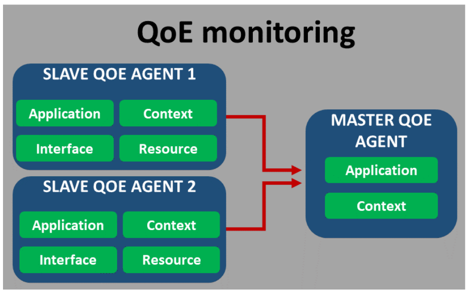


Figure 4.5.1: The task model which follows the QoE agent distributed approach for QoE monitoring

4.5.2 Network Model

The network is modeled as $G_N = (N, E_N)$. The vertices represent the nodes $N = \{1, ..., i, .., N\}$ and the links are described by the set of edges $E_N = (e_{ij})$. Each node is a Network Element (NE) which can be based on NFV where each NFV includes many VMs. The network model is presented in Figure 9.2.1.

The set of node types is described as $LT = \{1, ..., \delta, ...\Delta\}$. Each node $i \in N$ belongs to a set of locations based on its type, and I define that it can be on of the following, (i) Network Element, (b) MEC Element or (c) Cloud Element. I denote a node $i \in N$ with type $d \in LT$.

A task (a Master or Slave agent) $\lambda \in T$ is executed in a node $i \in N$ and the type of the node is $\delta \in LT$. Thus, I denoted a node as i_δ^λ.

A binary vector $X_i = [x_{i\lambda}]$ for $\lambda \in T$, can be assigned to each NFV-able node i in the network. $x_{i\lambda}$ is a boolean value which represents the state of the node i corresponding to task λ. When node i executes task $\lambda \in T$ then $x_{i\lambda} = 1$. Both of the tasks should be executed at least in one node in order to monitor the QoE level. Depending on the network parameters, the application type and the QoE estimation model, my goal is to decrease the complexity in terms of CPU and memory utilization, and the overhead by making the most appropriate placement of the QoE agents to network nodes.

The aim is to propose a lighter approach for QoE monitoring, considering the deployment location of the QoE agents. This is an NP-hard problem, which could be time and energy-consuming [94] if I have many services and huge traffic. I apply a centralized optimization approach to make the most efficient QoE agent placement in order to avoid the overhead increment during the QoE monitoring. Thus, I defined a utility function acc_{net} is assigned to a network for a given vector X_i. The goal of the network is to maximize its own utility. My solution chooses a QoE agent placement solution X_i that maximizes utility function. A strategy X_i^* is preferred to a strategy X_i if $acc_{net}(X_i^*) > acc_{net}(X_i)$.

4.5.3 Problem formulation

In order to formalize the correlation between the QoE agent performance, the network performance and user-perceived quality, a utility function is defined. My algorithm decides which particular NE should execute task $\lambda \in T$ by maximizing network utility function. The overall utility function consists of the cost of agent implementation in term of memory and CPU utilization and the QoE estimation for a node $i \in N$ executing a task $\lambda \in T$.

The objective function is defined as:

$$acc_{net} = max \sum_{i \in N} \sum_{\lambda \in T} (\alpha \times c_{i_\delta^\lambda} - \beta \times b_{i_\delta^\lambda}) \times x_{i\lambda} \tag{4.14}$$

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 (MQA, SQA) $\lambda \in T$. $c_{i_\delta^\lambda}$ refers to the execution cost for node i to run a task λ and its node type is δ. It is defined as the cost for resource consumption of both CPU and memory and can be calculated as follows:

$$c_{i_\delta^\lambda} = \gamma_i \times CPU_{i_\delta^\lambda} + \delta_i \times Mem_{i_\delta^\lambda}, \forall i \in N, \forall \lambda \in T, \forall \delta in LT \tag{4.15}$$

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 "Master QoE-Agent" needs more CPU and Mem (memory) than a "Slave QoE-Agent" task because it's going to conduct the QoE estimation using an estimation model. These weights depend on node type i_δ.

$b_{i_\delta^\lambda}$ is related to the execution benefit, if a task $\lambda \in T$ is executed in node $i \in N$ and its type is $\delta \in LT$. In the benefit $b_{i_\delta^\lambda}$ function, different QoE estimation models can be considered. However, in this work, I use a correlation model from [100] to map the QoS parameters to a QoE metric and a correlation model from [101], which correlates QoE and QoS metrics. Both of them are QoE estimation models for an adaptive video streaming service. I chose to use two models of the same type for the same service in order to observe if there is any impact on computational complexity during the QoE estimation.

A correlation model from [100] is used to map the QoS parameters to a QoE metric considering application parameters such as the video resolution, the Group of Picture (GoP) and the estimated video quality. This model is related to the QoS level considering the delay, jitter, bandwidth and packet loss.

The correlation model is derived by a normalized QoS value as follows:

$$b_{i_\delta{}^\lambda} = Q_r \times (1 - QoS(C))^{\frac{QoS(C)\times A}{R}}, \delta\forall \in LT, \lambda\forall T \tag{4.16}$$

where A is a constant related to the video resolution class such as Standard Definition (SD) ($A = 120$) or High Definition (HD) ($A = 240$). 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 [100], 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. 4.17. The $QoS(C)$ value can be calculated with the total sum of the values multiplying the measured QoS parameters in the network layer with the allocated weights. These weights are selected according to the type of the access network for the service. The considered QoS parameters are Packet Loss (PL), Packet Jitter (PJ), Packet Delay (PD) and Bandwidth (BW). The normalized QoS value reflects the network condition and is calculated as follows [100]:

$$QoS(C) = PL \times W_{PL} + PJ \times W_{PJ} + PD \times W_{PD} + B \times W_{BW} \tag{4.17}$$

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 is given from [100] 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.

An exponential mapping function is used to correlate the QoE metrics and QoS metrics [101].

$$QoS = K * \{(W * BW\} \tag{4.18}$$

$$QoE = a * e^{-\beta * QoS} + \gamma \tag{4.19}$$

Where BW is the bandwidth, K is the constant network factor (it depends on the network type) and W is a weight factor which defines the importance degree of QoS metric based on the application type (adaptive video streaming the the most important). α, β are network related parameters and γ is a video codec parameter. The result of the QoE model is a value between 0 and 1, which is normalized to represent user's satisfaction MOS (1 to 5). When the value is 0-0.2 the quality is bad and during the increment it become better. When the QoE is 0.8-1 the quality is excellent.

The objective function is subjected to the following constraints.

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

$$\sum_{i \in N} x_{i\lambda} \geq 1 \forall \lambda \in T \tag{4.20}$$

Constraint 2 Each QoE agent, depending on the application type requires a certain number of CPU cores for processing a unit. In case of an NFV-capable node, computational power is limited, and it will impact the assignment of QoE agent at that node. Each NE has specific available resources, and every task requires a specific amount of resources. Thus, the available resources for a NE cannot be less than the required amount of resources. For node $i \in N$ of type $\delta \in LT$, I define a set of available resources as $Available_i = \{CPU_i\delta, Memory_i\delta\}$. For a task $\lambda \in T$, I define a set of required resources as $Required_\lambda = \{CPU_\lambda, Memory_\lambda\}$.

$$Required_\lambda \leq Available_i\delta, \forall \lambda \in T, \forall i \in N, \forall \delta \in LT \tag{4.21}$$

Constraint 3 The $b_{i_\delta \lambda}$ is subjected to the service. Meaning that different QoE estimation model will be used based on the service type.

Chapter 5

Virtual Probes: Application Domains and Challenges

5.1 Introduction

Network service providers are facing the issue of the continuous increase in the demand of traffic and quality from their customers, which is mostly caused by the widespread deployment of multimedia bandwidth-hungry applications, such as video streaming. To flexibly address this issue, the providers are resorting to the new softwarization technologies, such as Software Defined Networks (SDN) and Network Function Virtualization (NFV). Thus, they need the Virtual Probes (vProbes) that are based on these technologies to probe the traffic because the current monitoring tools such as physical probes are not able to monitor a virtual environment. This work surveys the relevant techniques and the approaches that have followed towards vProbes' implementation. I mainly focus on the application domains and on requirements that they have to address. As the vProbes rely on the virtualization of functions and services, they inherit its advantages regarding scalability and flexibility, with a significant benefit concerning OPEX and CAPEX. Although, there are some open issues for the vProbe's deployment from the network providers such as the coexistence with the legacy networks and accuracy of the measurements.

5.2 vProbe description

Based on the approaches found in the literature, I define a vProbe as **"A network function in a virtualized environment with the ability to monitor intrusively or non-intrusively the traffic between physical or virtual network elements. It runs on a VM (as VNF), analyzes traffic, extracts useful insights and measures Key Performance Indicators (KPIs) that can be used to perform**

functionalities such as: providing visibility for network performance, user's QoE and detecting network problems.".

In the following, I describe the requirements of a vProbe and the functionalities that a vProbe has to provide. I start by analyzing which requirements are related to physical probes (indicated with a blue star in Figure 5.2.1), and which are typical only of vProbes (indicated with a red star in Figure 5.2.1).

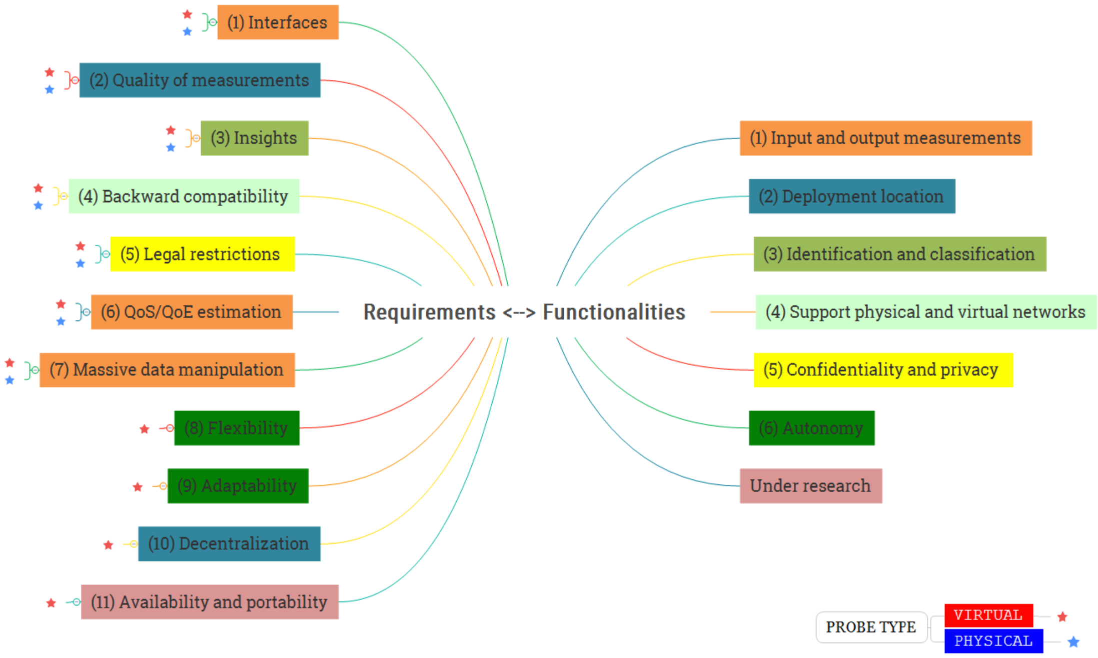


Figure 5.2.1: Requirements of vProbes and their related Functionalities

In each of the identified application domains, the **vProbes are preferred** to physical probes, because they are more **flexible, adaptable, distributed, portable and available**. The vProbes are more **flexible and versatile** than the physical probes because they can run instantly (e.g. as a VNF) in any location of the virtual network, to measure network parameters, to detect a network failure, or to monitor the level of satisfaction of user for a specific service [80]. The vProbes can be **distributed** in a virtual network depending on the needs of the provider and can act as centralized entities. Nevertheless, vProbe's components may be distributed in the network to perform different functions. For example, a component of a distributed vProbe can be responsible for making network measurements in one part of the network and another component in another part of the network, or if the vProbe had identified multiple network failures, one component might resolve a specific type of failure, and another component may determine a different kind of failure. Another essential characteristic of the vProbes is the **portability** because they can be moved from one virtual environment to another [102] [103]. Furthermore, in case a network administrator needs to update or extend the functionalities of a vProbe, this can be done on-the-fly without the need to shut down the vProbe [104]. Thus the vProbe remains **available**.

Regardless of all the advantages of the vProbes, there is still an **important requirement** that they cannot fulfil successfully, and this is the **accuracy**. The accuracy is related to network measurements, to detect a network failure or monitor and estimate user's satisfaction. The network measurements must be accurate, to estimate the network status with success and the failure of the detection must be accurate, to resolve any network issue with success. The problem with accuracy in virtual environments has attracted the attention of research and academic community. The virtualization technologies have an impact on several performance metrics of the protocol stack, and many factors might impact the accuracy of the results from monitoring /troubleshooting when virtual elements are involved. Moreover, **clock accuracy** is essential in case of latency measurements (i.e. Round Trip Time) and **time measurement accuracy** is important and affected by several factors such as VM scheduling or OS's rate limiters such as token bucket shapers [105].

5.2.1 The requirements of a vProbe

In the following, I present the requirements of vProbes, some of which are in common with the physical probes.

Common requirements with physical probes

Interfaces The input and output interfaces must be defined **based on the application domain** of the vProbe. Also, these interfaces should be open, to be able to flexibly specify which parameters to monitor in the network and forward the output to another network entity for further processing [64][66][68] [76] [78]. The existing SDN and NFV interfaces can be used after modification in order to enable a new Software Defined Monitoring control interface to transport the metadata and the packet flow data that are needed by the network monitoring applications. This data will be transported from the probes to the SDN Controller or from the switches to the SDN Controller [54].

Quality of measurements The quality of measurements is related to some factors that can influence it, such as the **location of the probe or the measurement's accuracy**. If the vProbe is installed near to the user's premises, it is possible to have a view of the performance close to what is experienced by the user. Thus, the quality of the measurements is considered higher. The vProbe must be placed in a proper location in the network regarding the measurements type [106], the measurements' accuracy [81], the resource consumption and the latency [107]. The optimal allocation of vProbes is related to many principles such as setup cost, economic profit, resilience, risk exposure, energy-related considerations, and acceptance rate [108]. Moreover, the accuracy of a vProbe is an important requirement and a challenge for any application domain because it has an impact on its functionalities. Note that, a vProbe is considered accurate when its measurements are accurate and close to the real values measurements. As an example, in a troubleshooting activity, if the vProbe is not accurate, this may lead to incorrect identification of the problem's cause. Furthermore, when performing measurements in a virtualized environment, significant errors may have occurred, because of the additional layers between the application and physical resources (e.g. time-related measurements impaired by the sharing of physical resources such as CPUs and data, buffers). Any application domain has its own methods (e.g. monitor the network, estimate QoE), and the accuracy is related to the performance of them. Some approaches in the literature refer that high accuracy is offered during measurements; however, they did not provide detailed information on how they supported it [84].

Insights (Analytics) The vProbe must be able to **identify and classify the network traffic** based on the application type. The identification of the network traffic will assist the network providers to extract useful insights, in order to manage their network efficiently and take better decisions [109][110][111][112][113].

Backwards compatibility The vProbe must be able to monitor **physical and virtual network elements**. In some cases, the network operators only support virtual networks such as in [65] [76] [67]. or virtual network in combination with physical networks. To facilitate the legacy mobile operator's transitioning to new virtual networks, the vProbes should have compabitiblity with transitional (physical) monitoring systems [54]. The commercial products that follow this approach are [71] [72] [73], [85] [77].

Legal restrictions A network provider when uses a vProbe must consider the legal restrictions. These are related to the **data placement**, where the data are going to be aggregated and the **technologies** that are going to be used preserving the **confidentiality and privacy** of users' data [114] [115] [116] [117].

QoE/QoS estimation The vProbe can collect data and extract useful insights, to estimate the QoS/QoE level [64][84]. **Parametric QoE estimation models** can be used for the estimation of the QoE level of service because they are the most appealing candidates for quantifying QoE levels in an indirect and user-transparent way [118]. Parametric planning models use network planning parameters and measure the values of specific network metrics. Different parameters are considered that are based on the service type. The vProbe can measure these parameters and estimate the QoE level.

Massive data manipulation Big data change the nature of data monitoring and management because it introduces new concerns about the volume and variety of corporate data. Thus, a vProbe needs to handle massive data inputs. The vProbe must have the required **computing resources**, to manipulate extensive data from input interfaces and suitable functions[119] [120].

Requirements related only to vProbes

Flexibility vProbe should be designed to allow for **dynamic instantiation and migration automatically** in a commodity server in an NFV environment without the need for new hardware appliance [30]. Thus, the network providers need to be able to apply a new vProbe instantly on multiple platforms.

Adaptability The virtualization can flexibly adjust VNFs placement depending on use-case requirements. A vProbe should be designed to be adaptable and to **change its behaviour without being intrusive** based on the application type and its requirements, the domain type and its requirements, computational and network load [121], and the topology [122]. It should have the ability to **tune its monitoring activities** considering the computing and communication resources in order to meet its goals. Furthermore, vProbes should be designed in such a way to **react fast** maintaining their precision. Nevertheless, this is still a challenge. The vProbes should be able to be **configured in real-time** depending on the workload, with new monitoring functions or by changing the existing monitoring methods [122].

Decentralization A vProbe should be deployed as a decentralised entity. This entity may include **many components**, and each component has a **different monitoring activity to execute** (e.g. to monitor different parts of a network). In a virtualized environment, these components can be **logically distributed** in the network (meaning that they are not in the same physical place, but they are logically connected) and controlled by a central entity decreasing the workload and increasing the management flexibility. This central entity is the main function of the vProbe, which is responsible for managing the different activities and process the data from them [61]. To support decentralization, the vProbe monitoring system could be **container-based or VM-based**. Containers can provide better agility and performance, as in this case, the VNF can run directly in a virtual environment. The monitoring components separated from vProbes instances, and the monitoring container/hypervisor can monitor multiple VNFs which belong to the same or different tenants.

Availability and portability The vProbes should offer availability and portability. They need to be moved to and deployed in different infrastructures with a **minimal effort**, and they need to be able to be **kept running** in case of maintenance or an upgrade [102] [103]. It is a significant difference concerning physical probes, which in similar situations need to be shut down. In the case of vProbes, it is possible to migrate them temporarily to another computer and migrate them back when the upgrade or maintenance completed.

5.2.2 The functionalities of a vProbe

Based on the requirements I mentioned in Section 5.2.1, I describe the main functionalities that must be applied, to fulfil them.

Input and output measurements A vProbe consists of two interfaces: the **input** interface, which is responsible for *capturing the network traffic*; the **output** interface, which is used to *forward the network traffic* to an "Aggregation point" for further processing and analysis using REST APIs.

The **input interface** typically captures traffic by using the "libpcap" [123] interface through the libpcap library. It performs PCAP packet capture and retrieval for off-line analysis using third-party tools such as Wireshark [124] or tcpdump [125]. Note that the packet analysis depends on the level of encryption that has a significant impact on the performance of a vProbe. Namely, increasing the level of encryption of service, the vProbe requires more processing time. The parameters that can be measured are Source/Dest IP address, port number, timestamp, signature, sequence number, flow application protocol, number of packets, session duration, delay, jitter, TCP handshake Round Trip Time and packet loss. Equally, to identify the type of flows, the vProbe can use the "ntop" [126] tool which is an open-source monitoring application and characterizes application protocols and identifies user traffic behaviour. Moreover, ntop used because it can provide a real-time view of the network traffic flowing in large networks and at the same time, provide useful analytics able to show KPIs and bottleneck cause analysis. The network flow classification can be conducted using DPI mechanisms [127]. Regarding the input data measurements, I note that the vProbe needs to collaborate with other VNFs, to handle massive data flows. To realize that, one solution is to maximize the scalability using parallelization of queries obtaining a highly scalable data streaming infrastructure [119]. That technique splits queries into sub-queries, which are allocated to independent virtual nodes in a cluster in such a way to minimize the distribution overhead. One solution could be the usage of stream processing engines [120]. Each stream processing block in the engine includes a single stream query and can be combined with other blocks to obtain complex elaboration chains. Stream processing blocks are application-independent and re-used in almost any processing task. Regarding chains, they are defined based on the application type, and they enable complex filtering, elaboration and aggregation of data flows. NFVs are used to deploy the stream processing block. This block is a component that takes one or more event streams as input and generates one or more event streams as output. The output and input streams correlated through a processing function. Thus, the vProbe, in collaboration with other VNFs, can manipulate massive data flows and monitor the quality at the same time.

Regarding the **output interface**, through it, the relevant data are sent to the "Aggregation Point" such as an Aggregation Server, where the network administrators correlate them with the other network traffic information. It is possible to export data from ntop using the log files. Export in log files performed through scripts that can access the monitoring engine and dump data into log files.

Deployment location The deployment of probes is related to the issue of deciding *where to place them* and *how many* are sufficient to cover the network. The placement has been the subject of several research works [128][129] [81]. With NFV, vProbes can be instantiated in any convenient location [130]. Nevertheless, they must be placed in a proper location considering the *measurement's type*. There are three different perspectives [106] [131][81] related to measurement's type:

1. when a vProbe is at the ISP's side in a location such as data-center, it is going to measure bandwidth utilization, packet loss, round trip time, and other network-related parameters,

2. when a vProbe is at the vendor's side, it is used to trace samples and analyze logs;

3. when a vProbe is at the user's side, it is going to measure the bandwidth availability, response time or packet loss.

The *VNF placement* is important, and it is related to the performance and the efficiency of a network. Integer Linear Programming with cost constraints for VNF can be applied to deal with the probe placement issue [132]. The appropriate placement depends on the **operation cost [78], the number of probes and the functional targets**. Moreover, this problem is related to **resource consumption** (physical or virtual resources) and **end-to-end latencies** [107]. Thus vProbe as a VNF inherits this problem. An option to solve this problem is the deployment of vProbes in a cloud-based environment owned by a network provider. This may have some advantages such as cost restriction, flexible updates, dynamic migration and management of the entire infrastructure. Currently, there are just a few operators that use NFV environment, so the traffic overhead is not critical for the network. However, an increase in traffic exchanged between cloud-operators and network operators are expected [108], and this could be challenging when vProbes deployed in a cloud environment.

Identification and classification Traffic classification is an important part of network monitoring. Information can be extracted from encrypted or non-encrypted connections, mainly from the session's initiation. However, before the analysis and the classification, the network traffic needs to be identified. Application identification methods are *statistical or behaviour-based* [133].

The network traffic can be classified as inflows [134]. The captured network traffic creates a collection of flows. When network interfaces receive raw packets, there is a need for inspection of the packet payload. Using the 6-tuple (VLAN, IP source, IP destination, port source, port destination, protocol), the ntop tool attempts to detect the real application protocol carried by the flow. ntop discovers the application protocol using the nDPI [135] library, which recognizes 170 protocols and based on a protocol-independent engine that implements services common to all protocols, and protocol-specific dissectors that analyze all the supported protocols. Thus, the vProbe is responsible for identifying the flow using the ntop tool and forward the results for the type of application to the "Aggregation server". However, ntop is not able to identify the traffic type when it is encrypted. Nowadays, most of the content providers apply application-level encryption to deliver their services with privacy to the end-users [53]. QoE monitoring or network monitoring using vProbes needs to be evolved and *be independent* of the application layer [64]. New methods are needed to calculate the QoE level efficiently through transport layer inspection. HTTPS is the most used protocol for secure communications over the Internet, but an issue that arises is the identification of HTTPS traffic without relying on decryption [136]. Legacy solutions, such as Port-based and DNS, could be used to solve this issue in legacy networks, but they do not work in virtual environments. Decryption methods, such as HTTPS proxy or crack encryption algorithm could be used, but they increase computation complexity and raise privacy issues raised. Thus, this issue could be solved using a flow-based statistical method, which applied to encrypted traffic. Nevertheless, the low accuracy and computation overhead issues still exist. Using an improved flow-based analytical method that identifies the HTTPS services in service-level without relying on header fields and without decrypting the encrypted traffic could solve this issue [136].

Support physical and virtual networks A vProbe needs to collaborate with physical probes in legacy systems [78][77]. The physical probes (in physical devices) could be attached to an aggregation network which includes the vProbe or close to the end-user [77] [72]. An orchestration strategy needs to be considered, to *close the gap between legacy services and NFV.* It is essential to maintain a migration path; namely, the route followed while changing from the use of one type of network to the use of a different one toward NFV while keeping operators' current network investments in place [137] In order to support the backwards compatibility, the vProbes should be able to analyze the different user and control plane traffic flows using new or existing interfaces between the SDN and the existing networks [54].

Confidentiality and privacy An essential functionality of vProbes is to handle data so that legal restrictions are followed. Legal restrictions are related to data placement: where the data are aggregated and which technologies used, to *preserve the confidentiality and privacy* of the user's data. Thus, secure techniques must be used at the server's side, such as cryptography. The network providers need to comply with the *privacy legislation* and adopt some *data management practices based on privacy policies* to keep their credibility in front of their customers [115]. As I discussed earlier, confidentiality depends on the collection and processing of information; thus, issues related to personal data protection are raised. Nevertheless, the most important fact is that the virtual environment is cloud-based [138]. Privacy implications are present; thus, there is a need for adequate access control for privacy. Therefore, regardless of the application domain, a vProbe in a virtual environment **must be privacy-aware** and **follow a policy-based access control model**, to enhance the particular application domain with privacy-awareness [114].

Autonomy The vProbes can be deployed **dynamically** and **autonomously** considering at the same time that they should use the minimum storage and the minimum communication overhead [139]. Moreover, based on non-cooperative game theory, the vProbes' *can negotiate with each other* to decide which one is going to execute each function of a vProbe. Following that approach, a vProbe can have autonomous behaviour, decrease the computational complexity and the average energy consumption [140].

5.3 vProbes in a NFV-based network architecture: An Example

Based on the considerations of the previous sections, in the current section, I present a possible NFV-based network architecture using vProbes. The vProbe-based architecture consists of 3 parts: the physical environment, the virtual environment and the aggregation server, as shown in Figure 5.3.1. In this Figure, the elements of a vProbe-based architecture are depicted and associated with one or more

"Physical environment" It includes the physical network, the storage and the computation power of the infrastructure of the network provider. This is the physical part of the infrastructure. The virtual resources are instantiated on these physical resources. Any commodity switches, servers or storage servers belong to this category [6].

Note that, traditional physical resources, which have no connection with virtualization technologies, should be able to collaborate with the virtual resources (4. Functionality: Support physical and virtual networks).

"Virtual environment" The virtualization environment includes the "Virtualization layer", which is the Hypervisor. The Hypervisor runs and manages physical resources; and executes the VMs. Some examples of VM deployments are: Docker [141], VirtualBox [142]. Through the Hypervisor, the network administrator is able to organize the vProbes properly. Some examples of Hypervisor deployments are: KVM [143], XEN [144], ESXi [145].

For every application domain, the vProbe collects the related parameters (namely, the inputs) and exports the relevant results (namely, the outputs) (1. Functionality: Input and output measurements). The location of the vProbe is defined by the network administrator based on networks needs (2. Functionality: Deployment location). The vProbe can identify and classify the incoming traffic using the appropriate technologies (3.Functionality: Identification and Classification). Each vProbe can cooperate with the others presenting an autonomous behaviour (6. Functionality: Autonomy).

I extend the NFV architecture and propose the usage of an external entity called "Aggregation server". The "Aggregation server" includes a platform that provides information to the network provider to better understand the service quality that the users are experiencing using their network or detecting unpredictable issues. This network part integrates, associates and processes data from the vProbes, in order to meet its goal based on the particular application domain. According to the obtained results, the network provider will be able to improve and optimize its network. Nevertheless, the network providers could use their platform (if it exists) or use existing open-source solutions such as ElasticSearch [146], Solr [147], Lumify [148], Joojip [149] or commercial products by companies such as Huawei [150]. Note that, this platform must have the ability to store the results considering the privacy and the confidentiality of the users (5. Functionality: Confidentiality and privacy) and include effective web interfaces to visualize the results.

5.3.1 Challenges and lessons learned

In the current section and in Figure 5.3.2, I identify some challenges related to the deployment of vProbes by a network provider and the proposed solutions.

5.3.2 Virtualization

The challenge: A vProbe inherits many issues from *virtualization technologies* related to resource isolation, lack of visibility, portability, problem detection, and interoperability.

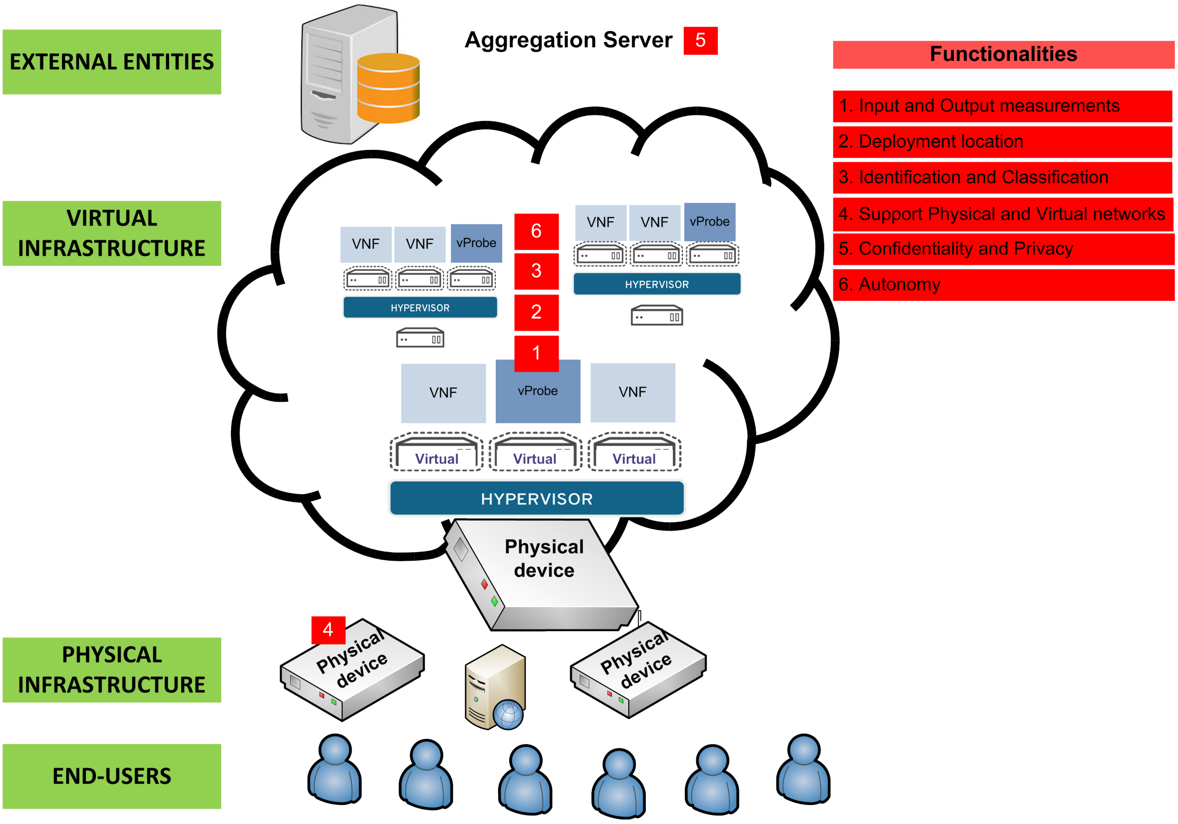


Figure 5.3.1: vProbe-based NFV architecture and its related functionalities

Some vProbes as VNFs are executed directly on a bare metal server, which is a single-tenant physical server, ensuring predictable performance, because the mapping of software instances to hardware instances is predictable. Nevertheless, this deployment leads to **resource isolation**, and it is difficult to secure the software instances because of multiple software appliances executed as processes on the same Operating System (OS). This deployment enables each VNF to be performed on its specific OS while remaining unaware of the underlying OS. Moreover, another issue that is inherited by the virtualization technology to the vProbes is the possible **lack of visibility** of network traffic from network interfaces [83], because these interfaces are virtualized. The vProbe's **portability** across the underlying infrastructure is another issue, and its configuration should be done automatically and not manually. Moreover, a vProbe needs to specify if a **detected problem** came from itself or a collaborative VNF and then apply the appropriate functions to resolve it. Also, in many cases, the hardware and the software components are provided by different vendors; thus, there are **interoperability** issues.

The solution: The VNF resource isolation ensured because VNFs executed on independent VMs managed by the **hypervisor layer**, which guarantees no unexpected interactions between them [6]. **OPNFV**[151] can deal with the lack of visibility using a mechanism to source traffic from/to a physical probe to/from a vProbe, and with fault and failure detection in the NFVI, VIM and other components of infrastructure and recover from them. Moreover, OPNFV forms an open environment for vProbes based on open standards and open-source software to provide interoperability. **Open Virtualization Format (OVF)** [152] can address the portability and deployment of physical probes and vProbes. It enables the packaging and the secure distribution of vProbes, while at the same time provides cross-platform portability and simplified deployment across multiple platforms. The NVF environment of the vProbe should be based on a cross-platform virtual resource manager to ensure its portability [6]. To address aspects related to the performance and the portability of a vProbe, the **Enhanced Platform Awareness (EPA)** [153] principles can be applied as happened with OpenMANO [154] project [155]. EPA can facilitate better-informed decision making related to vProbe placement and help drive tangible improvements for NFV tenants [156]. Furthermore, **ETSI** has created a document about the "NFV Performance and Portability" [157] that explains how to apply the best practices related to performance and portability in a broad VNF deployment.

5.3.3 Security

The challenge: Security threats analyzed in the ETSI recommendation [158] and most of them considered in the case of the vProbes. As NFV gets deployed and more essential functions virtualized, I can expect it to attract even more security and privacy threats. More than ever, there will be threats based on data interception (whether lawful or otherwise) [6]. If the network provider does not receive any countermeasures against security threats, it could lose: i) its **confidentiality** by eavesdropping attacks, ii) its **integrity** by traffic/data modification from hackers or iii) its network control by attackers who can compromise network elements using unauthorized access with the risk of **data leakage**. Security must be applied in all the NFV domains in the hypervisor, in the computing domain, in the application domain and the infrastructure [6]. The hypervisor domain deals with two security challenges that are unauthorized access and data leakage. The computing domains deal with the shared computing resources (e.g. CPU and memory) and their security challenges. The network domain deals with the divided logical network layers (e.g. the virtual switches) and shared physical network interface controllers.

The solution: In order to keep the NFV environment and vProbes deployment secure, some practices that can be applied are: separating the vProbes in different **trust zones** like traditional firewall zones, protecting hypervisors using **hardening techniques** such as disable unused ports or **monitoring integrity critical files** and **protecting the network** using mechanisms such as DHCP snooping and ARP inspection [75]. Considering all the domains the solutions that can be followed are: for the hypervisor domain is to apply authentication controls only by VMs, for the compute domain the solution is to encrypt the data and get accessed only by the VNFs and for the network domain the solution is to adopt secure networking techniques such as SSH and IPSec [6]. To deal with privacy and confidentiality violation, the countermeasures that can be utilized is to use **cryptographic techniques** such as Trusted Platform Module (TPM) [159]. Moreover, **ETSI** has created a "Security Guidance" that guides how security, privacy and trust may be achieved in NFV environments [160].

5.3.4 Computing performance

The challenge: Network providers must deliver their services with high quality to their subscribers, to reduce users' churn and increase their profits. Therefore, the **performance of the physical equipment** in the virtual environment of a vProbe has an important impact on vProbe's performance as well as the vProbe's **design**. In the perspective of the users, the services whether based on functions running on dedicated equipment or VMs should have the same performance. Thus, a virtualized environment and a vProbe must achieve the same level of monitoring, like the traditional network environments and the traditional/physical probes [6], and the service levels must be granted without degradation during appliance load and relocation [161]. The vProbe's **availability** requirements for NFV should be at least the same with these for legacy systems for the same service. Concerning the requirements' fulfilment, the NFV components need to provide the same or better performance in one or more of the following aspects: failure rate, detection time, restoration time, the success rate of the detection and restoration, impact per failure, etc.[162].

The solution: However, the achievement of the same level of monitoring, like the traditional network environments is complex and is not easy to take place [163] due to the integration of open-source software, COTS hardware and software components in a system. The design techniques that deal with the performance issue are under research and include [6]:**multi-threading techniques, independent memory structures to avoid OS deadlocks, closed network stacks, methods for direct access to interfaces to increase data throughput and reduce latency**. To meet service availability requirements, the vProbe design needs to take into account factors which include **commodity-grade hardware** and the presence of multiple software layers (i.e. hypervisor and guest Operating System) [162]. The carrier-grade deployment of vProbes requires a lightweight VM implementation such as ClickOS [164] and the development of other NFV services, to monitor the performance of the vProbe itself. As I mentioned, some functions in industry-standard servers may not achieve the required levels of performance; thus, **OpenANFV**[165] proposes an OpenStack-based framework, which uses hardware acceleration to enhance the performance of VNFs. It aims to provide automate and elastic provisioning for hardware acceleration to VNFs in OpenStack.

5.3.5 Coexistence with legacy networks

The challenge: Another major issue, for network carriers, is the migration from the traditional network infrastructure to NFV-based infrastructure, given the **massive scale** of the traditional networks and the **firm attachment between its components**. Thus, the vProbes should be able to coexist with legacy network equipment and be able to interact with legacy systems with minimal effects on existing networks. Nevertheless, they must ensure a **secured transition** from physical functions to VNFs, without interrupting the service or impacting on the network performance [166] [162].

The solution: An **API could be defined**, to interact with legacy systems with minimal effects on existing networks [6], [167]. The **NFV architecture** needs to enable a **hybrid network** to support physical probes and vProbes, and it should provide a smooth path of migration from the physical network to an open standard based virtual network [168].

5.3.6 Accuracy

The challenge: Another major issue that must be handled by the vProbes is the *accuracy* of their measurements because with vProbes can occur more **delays** and higher possibilities for **errors**. Some of the approaches from the literature have mentioned that they deal with this issue, but there is no explanation for how they deal with that [84] [68].

The solution: Based on [81], the vProbe's **location** is an essential characteristic of it and its geographical placement affects its accuracy. Thus, if the vProbe deployed at the edge network close to the user, it has higher accuracy. Measurements' accuracy is still an obstacle for the adoption of vProbes by network operators.

5.4 Conclusion

In this work, I have surveyed the works on the vProbes as a VNFs, used to monitor a network, detect unexpected network issues and monitor the QoE level using the technologies mentioned above. Thus, considering this concept, I have identified and described vProbe-based approaches, and I have divided them into four application domains: network monitoring, QoE monitoring, troubleshooting and other cases. I have observed that vProbes are preferred in comparison with physical probes because they are more flexible, adaptable and cheaper. Nevertheless, the hybrid approaches are the optimal solution to deploy a vProbe, because most of the network providers are under a “virtualization technology” adoption phase and in hybrid approaches the

Challenge	In terms of	Solutions
Virtualization	• Resource isolation • Lack of visibility • Portability • Problem detection • Interoperability	• OPNFV • OVF • EPA principles • ETSI "NFV performance and Portability"
Security	• Confidentiality • Integrity • Data leakage	• Trust zones • Hardening techniques • Monitor integrity critical files • Cryptographic techniques • Authentication controls • ETSI "Security Guidance"
Computing performance	• Performance of physical equipment • vProbe's design • Availability	• Design techniques: Multi-threading techniques, independent memory structures, closed network stacks, direct access to interfaces • Commodity-grade hardware • OpenANFV
Coexistence with legacy networks	• Massive scale of legacy networks • Firm attachment between network components • Secure transition of the functions	• APIs should be defined • NFV architecture enable hybrid network
Accuracy	• Delays • Errors	• Location of the vProbe

Figure 5.3.2: Challenges and possible solutions

physical probes can be used in cooperation with vProbes. I have identified the common requirements of physical and vProbes and the individual requirements of vProbes and their corresponding functionalities. Based on these considerations, I propose an NFV-based network architecture using vProbes, its main components and vProbe's functionalities. In conclusion, considering the vProbe's characteristics and the current technologies used by network providers, I observe that there are some open challenges for the vProbe deployment which include: the existing challenges of virtualization technologies, the security threats, the performance of the deployment environment, the collaboration with the existing (physical) infrastructures and the accuracy of vProbes' measurements. The vProbe is still on early stage, and its performance and accuracy need more research efforts to be commercialized.

Chapter 6

A QoE monitoring and management solution

6.1 Introduction

In this work, I illustrate a Software Defined Network (SDN)-based architecture for Quality of Experience (QoE) management that solves two of the major problems of current networking technologies which are related to the limitations in scalability and flexibility. Its advantage is the exploitation of the virtualization features of the network nodes and devices to flexibly deploy monitoring and control functions in the different points of the network according to the SDN control functions. As a result, the QoE monitoring and management is deployed at the application layer on top of the controller. In order to evaluate the proposed framework and architecture, a platform has been developed, which is called QoE-MoMa (QoE-Monitoring and Management) platform, making use of the Opendaylight solution and Mininet emulation environment. To evaluate QoE-MoMa, I focused on the video streaming service. The final quality has been evaluated using the estimated MOS (eMOS) model that considers rebuffering events, duration of the rebuffering, switch quality rates, video resolution, and a quantization parameter. The results show the efficiency of the proposed approach observing that higher QoE level is achieved if I consider the application and network parameters. In conclusion, I consider that QoE-MoMa is useful as a QoE monitoring and management tool for a variety of services and can be deployed on a real network conveniently.

6.2 Related Works

The main goal of next-generation networks is to use the network resources efficiently and to optimize the traffic flow in order to succeed in high levels of QoE. To this end, better insights are required (e.g. real-time data collection), to make better decisions based on these insights (e.g. deep learning analytics) and act [169] accordingly. QoE is essential for telecommunication operators and service providers that are involved in the service delivery chain. It will be beneficial for operators to know the type of information, such as the application type, in order to enhance or adjust QoE levels.

A novel architecture component-based on new 5G network technologies is presented in [170] which solves the issue of QoE-awareness. A heuristic and QoE-aware user association algorithm are proposed in [171] which maximizes the network operators' profit and provides high QoE using a QoE-based charging scheme. In [172], the authors proposed a QoE measurement system, and virtual clients were used in order to generate traffic loads on different network devices in an independent way. The proposed system is not able to predict a user‘s behaviour in a real-life scenario. QoE is expected to measure end-to-end connection and applications running over a different kind of networks. In [173] the authors propose a high-level context-aware QoE management framework which consists of an Intelligent QoE entity and proposes the concept of "experience packages" which are based on user‘s profile and real communication scenarios. The authors in [174] proposed a QoE measurement framework that does not require user‘s participation in the process. Their framework can assess the impact of changes in the network in a video streaming service with accuracy and providing a closed loop. With the capability of a closed-loop, the QoE-aware service optimization can be done at runtime. The two QoE metrics that are measured are video quality and video switches.

The main difference with other approaches is that these QoE measurements are provided as-a-Service to third-party applications or network elements. The authors in [175] present a new approach based on bandwidth management and their aim to optimize the overall QoE of multiple video streams. An SDN-based architecture is presented including a "Video QoE optimization application (VQOA)" in order to monitor the video quality by collecting and analyzing information from the network. Then, in order to improve the video quality, it uses a dynamic bandwidth allocation algorithm based on the service type, the complexity of video content or business policies. The quality metrics that are used in the proposed framework are Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Mean Opinion Score (MOS). The authors in [176] proposed an SDN Controller for multimedia transmission intending to meet 5G Key Performance Indicators (KPIs) by maintaining the reliability and integrity of the network while and reducing latency in order to meet users' QoE expectations. The proposed scheme considering QoE metrics such as content type and QoS metrics explore scalable video encoding and provides dynamic tools for end-to-end multimedia management. An SDN-assisted QoE control method for DASH over HTTP3 is proposed in [177]. It comprises a multi-layer collaborative optimization at the use, application and network levels. Moreover, it combines the newest technologies of quality adaptation. This proposed method considers end-users categories to manage the video segment's traffic over QUIC, and the result was an overall QoE improvement in terms of media throughput, number of stalls, and the downshifts of video quality in comparison with TCP-based approaches. An SDN-based architecture that enhances QoE monitoring and management by integrating SDN and enhanced Telecom Operations Map (eTOM) is proposed in[63]. This architecture was deployed in case of a Service-Level Agreement verification in order to identify SLAs violation. It succeeds, and its mechanisms can adjust with a dynamic way the network parameters in order to satisfy the SLA constraints.

6.3 The proposed solution

6.3.1 A Framework for QoE monitoring and management

Currently, there are many services (e.g. video) offered by service providers. Nevertheless, users' demand for new and different services continue to increase. Thus, the providers are trying to meet these demands by continually purchasing, storing and operating new physical equipment in order to support them. Nonetheless, this is going to increase the OPEX/CAPEX and eventually, the service price by the service provider, which may, however, negatively affect the customer churn. In order to avoid this problem, they have to find another way to build more dynamic networks to reduce OPEX/CAPEX expenses and at the same time improve the Quality of Service (QoS) and QoE. QoE is introduced to consider the user's perception and satisfaction. With QoE-aware monitoring and management of their network, service providers can achieve their primary goal, namely to make profits and avoid users' churn. Using SDN technology, they can achieve that by changing the limitations of current network infrastructure. Accordingly, network features and services can be added dynamically under the form of network applications, which facilitates their deployment. Therefore, motivated by the possibility to offer a successful multimedia experience to users, service providers are encouraged to consider the degree of satisfaction achieved by the users who are paying for the service.

While service is begin executed, many subjective factors may influence the QoE level. In order to keep the user satisfied, the operator should consider user's requirements accessibility (access the service anytime), acceptability (the service worthies the paid money) and reliability (reliable communication). These requirements need to be fulfilled using any network technology and across any operator domain. Different applications may have different requirements in terms of QoE. In this solution, the requirements are linked to Influence Factors (IFs) and considered to be any characteristic of a user, system, service/application or context whose actual state or setting may influence the QoE of the user [3]. According to it, in my proposed solution, I retrieved information related to each IF in order to develop an efficient QoE monitoring and management system. Thus, I take into account user's characteristics (e.g. preferences), content's characteristics (e.g. content type), network's characteristics (e.g. data transmission over the network) and device's characteristics (e.g. system specification) and the factors that describe the user's environment (e.g. location).

As a first contribution, I aimed to propose a framework to organize the QoE monitoring and management functions as seen in Figure 6.3.1, and apply this framework in an SDN-based network architecture. I organized the significant functions in three layers as described in the following.

The **Data Acquisition Layer** is responsible for collecting data from the network and the users with reference to the System IFs, Human IFs and Context IFs through Probes (System IF), Deep Packet Inspection (DPI) (System IF), External services (Context IF) and User Profile (Human IF).

1. To retrieve some information about System IFs such as content type, I use the "Probe" component, which includes passive probes (virtual or physical). These probes retrieve network-related information. In the case of a virtual environment, virtual passive probes (vProbes) are used, otherwise traditional passive probes are used. A vProbe is a software or a function in a virtualized environment, and it is installed in a Virtual Machine (VM). It retrieves information from the virtual network and forwards them to the "Network Monitoring" component in the "Monitoring Layer". In case of a legacy network, passive physical probes installed in dedicated hardware devices inside the network, are used to forward the retrieved data to "Network Monitoring". I propose the usage of vProbes because they are more scalable than physical probes. Moreover, in order to install a new vProbe, I need to add another instance of the VM in the virtualized environment.

   To retrieve some other information regarding the System IFs, such as network-related parameters, the "DPI" component is used. Firstly, I identify and classify network traffic. However, nowadays in most of the cases, the traffic in a network is encrypted, thus in order identify the application type I use a "Payload Inspection" technique [178] and more precisely the DPI method [135]. Using DPI, I can identify the application type (e.g. browsing), the application protocol (e.g. HTTP or HTTPs), the application software (e.g. mail client software) and fine-grained services (e.g. skype voice call). If there is a virtualized environment, the DPI is installed on a VM; otherwise, it is installed in a dedicated hardware device. The network administrator can observe the results of DPI using a web interface. The results from the DPI method are forwarded to the "Traffic Classification" component in the "Monitoring Layer".

2. For the identification of Human IFs by the "User Profile" component, I need to retrieve user preferences which are obtained by the user's application. The

user defines his preferences, and I retrieve them using an API, and then I create the user's profile with his preferences for the service in question. I store user‘s preferences to the “User Context” component in the “Monitoring Layer”. This information is forwarded from the “User Profile” component to the “Context” component via an API.

3. To identify Context IFs such as user‘s location, the “External services” component is used which retrieves information about user's context using an API with the external service. The results are forwarded to the “User context” component in the “Monitoring Layer” via an API.

The **Monitoring Layer** retrieves information regarding network status, service type and stores information retrieved from “Data acquisition Layer”.

1. The “QoE Aggregation Database” component which is used as database and includes three sub-components. The “Network monitoring” component includes the network status using network parameters (e.g. bandwidth) and information retrieved from the “Probe” component in Data Acquisition Layer. The “Traffic Classification” component includes information from the “DPI” component. The “Context” component includes temporal information about the service in question such as the frequency of usage of a service, economic context (e.g. subscription type) and information retrieved from the “User Profile” component in the “Data Acquisition Layer”.

2. The “Content” component includes information about the content type (e.g. 2D or 3D).

3. The “System” component includes information about the media parameters (e.g. resolution) of the service in question.

The **Management Layer** is responsible for QoE evaluation and management. Namely, it decides which control operations should be executed afterwards to assure the best QoE to the users based on the information from the “Data Acquisition Layer” and “Monitoring Layer”.

It includes two sub-components:

1. The “Evaluation” component includes a sub-component called “QoE Model” that applies the most appropriate QoE model based on the type of the service (e.g. video). Different QoE models are used because different applications require different levels of network resources.

2. The “Management” component includes 2 sub-components: The “Control Algorithm” component includes an algorithm that implemented in the network in

order to improve the QoE level. The “Service Level Agreement (SLA)” component includes information about the contract between the service provider and the customer.

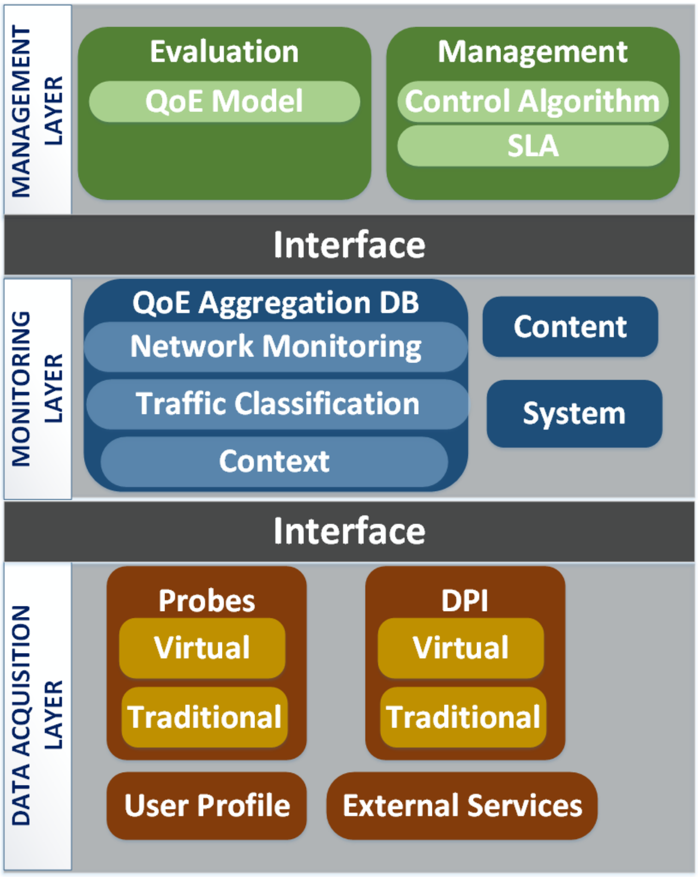


Figure 6.3.1: QoE Monitoring and Management Framework

6.3.2 SDN-based architecture

In this section, I propose an SDN-based architecture that implements the QoE monitoring and management components as described in Section 6.3.1. The result of this solution is shown in Figure 6.3.2.

The SDN Data Plane (DP) includes networking devices (e.g. switches) which are software-based (e.g. virtual switches) and have the capability to forward data packets and communicate with the SDN Controller[179]. In DP, I monitor the traffic in the SDN architecture using network monitoring probes. Passive vProbes are used because they present higher operational efficiency than physical probes, they are dynamically transferable across the network, and they reduce the cost instead of using passive physical probes. Furthermore, DPI is used in order to identify and classify network traffic. Virtual DPI is installed in a VM. It collects traffic and forwards the results to the “Network monitoring” component. The benefit of vProbes’ usage is the flexibility provided by the fact that they can be placed anywhere on the network efficiently (in a virtualized environment). Note that the network data from the users' devices are anonymized due to privacy issues.

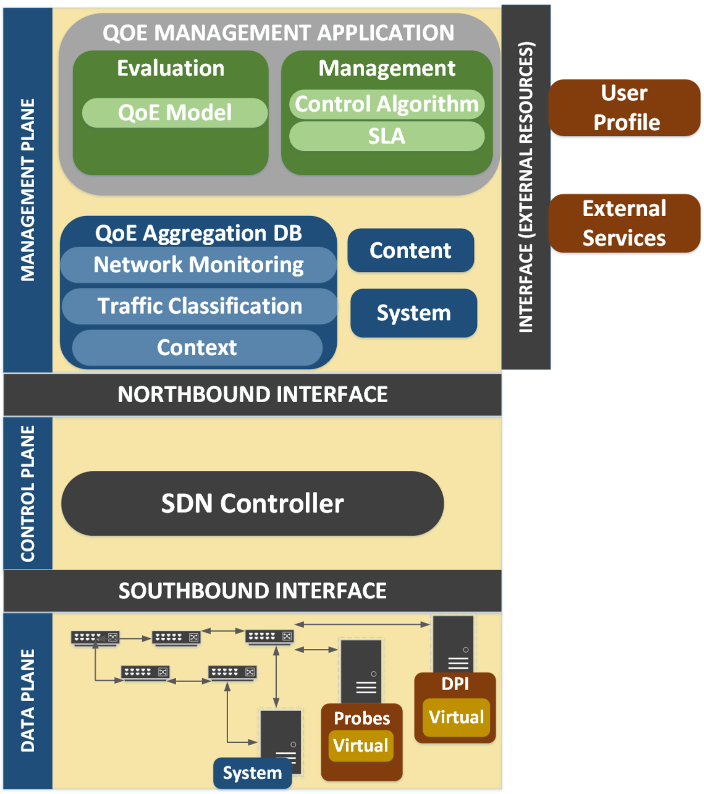


Figure 6.3.2: QoE Monitoring and management SDN-based Architecture.

The main component of the Control Plane (CP) is the SDN Controller. This Controller is software-based and has the responsibility to monitor and control the underlying network knowing all the available resources and the network topology [179]. In the proposed architecture, an SDN-controller communicates with the network in DP via Southbound Interface (SBI) using the OpenFlow protocol [27] and communicates with the above layer via Northbound Interface (NBI) using an information exchange protocol such as REST.
The MP includes the SDN application which interacts with the underlying network via SBI through the Controller. This application is executed inside the Controller and can configure the network devices[179]. In this work, I proposed an application for "Management" which includes five internal components and two external components.

The internal components are described as follows:

1. The "Evaluation" component retrieves information from the "Content" and "Traffic classification" components regarding the service type. Based on this, it implements the appropriate "QoE model" in order to evaluate the QoE level of the service.

2. The "Management" component implements a control algorithm based on the results

from the "Evaluation" component and "SLAs" component. It takes decisions to improve the QoE for the considered service.

3. The "QoE aggregation database" includes the following components:

   (a) The "Traffic Classification" component is related to traffic identification and classification. The technology used in this component is characterized by its' input, output and the technique that is used [178]. The traffic characteristics that are used for the classification as the input are the *traffic flows*[180]. Furthermore, the core classification method is related to the applied technique, and the output describes how the traffic will be mapped to traffic classes. I use DPI technique and more precisely, the nDPI [135], which is a flow-based tool that inspects traffic payload (supports encrypted traffic). DPI examines the first N bytes of the packet based on the syntax of packet's payload for each service. The desirable output is the type of application, such as a game, and the protocol type such as HTTP. Note that, knowing the type of the application will be easier to choose the most appropriate "QoE model" in the " Evaluation" component to estimate the QoE level and the most appropriate "Control algorithm" in the "QoE Manager" component to improve the QoE.

   (b) The "Network Monitoring" component includes information from the "Network state Information" module by the SDN Controller and helps to estimate the network's status. Also, it includes information gathered from the vProbes.

   (c) The "Context" component stores information related to the "User Preferences" for the service in question retrieved by the Over-The-Top (OTT) service (if there is a collaboration between OTT and ISP or preferences submitted by the client).

4. The "Content" component includes information related to application parameters for the service in question, e.g. for a video service, and it includes information about the video type.

5. The "System" component includes information related to the media-related parameter (e.g. resolution).

The external components are described as follows:

1. The "User Profile" component includes information about user's preferences for the service in question. This information can be defined by the user (using an agent installed in this device) or retrieve information from the content provider (if there is a collaboration between the content provider and network provider).

2. The "External Services" component retrieves information from third party services in order to identify the user's context such as geo-location services in order to identify user's location [181].

6.3.3 QoE-MoMa platform

In order to evaluate the performance of the proposed strategy, I developed "QoE-MoMa", a platform in which I applied my proposed strategy (i.e. the framework and the architecture) in case of adaptive video streaming application. ISPs or OTTs can use the proposed strategy and the platform in the future in order to monitor and manage the QoE levels in a cooperative way. Furthermore, it could be used for different types of applications, such as video games. In this work, the proposed platform is used only with one purpose: to monitor the QoE in case of an adaptive video streaming service in order to estimate user's satisfaction.

QoE-MoMa monitors the network QoS parameters using vProbes. The vProbe is installed in an OpenFlow switch in order to passively monitor KPIs of video streaming from the network's side. QoE-MoMa uses the QoE-Model from Section 4.1.

6.4 Performance evaluation

6.4.1 Experimental Setup

In this Section, the QoE-MoMa performance is evaluated by means of Mininet[182] and OpenDaylight[183]. A media server is used to stream video in Mininet. Mininet is an emulation environment which emulates a variety of network elements, including Layer-2 Switches. The media server and the target host are both inside Mininet and use VLC player version 3.0 (which is not yet stable but it supports MPEG-DASH streaming). I use network monitoring tools such as Wireshark [124] and iperf to monitor the network and then identify bit rate variations, bandwidth, throughput and latency of the network. Mininet was installed in a Toshiba computer with Intel Core i5 CPU @ 3.40 GHz, 6GB RAM installed with Linux Ubuntu 14.04, 32bit. Opendaylight is an open-source controller which includes support for the OpenFlow [27] protocol, among others. This controller is implemented in software and is not dependent on particular hardware or operating system. OpenFlow as an open communication protocol allows the controller to configure the network devices in DP. Figure 6.4.1, shows my experimentation setup. I deployed DASH-servers and video clients on virtual machines in a virtualized environment. It includes 2 OpenFlow switches, six hosts and 1 of them acts as Media Server, which is used for video streaming. I used "Big Buck Bunny" (BBB)[99] and "Elephants Dream" (ED)[99] video sequences, considering two different resolutions, high labeled as "H" (720p) and low labeled as "L" (360p). I have created a dataset of media parameters for test sequences, as presented in Table 6.4.1. In these sequences, I are considering the video bitrate (VB), the frame rate (FR) and the quantization parameter (QP). In some cases, I am changing the QP and in other cases, the video bit rate, and then I estimate the eMOS according to Equation 4.1. Note that I consider that rebuffering frequency is more important than the other coefficients. Thus I set to it a higher value. Accordingly, the coefficients for rebuffering frequency, quality switch rate and rebuffering duration are set to $0.50, 0.25, 0.25$, respectively.

To further evaluate the performance of my framework and how it captures QoE degradation caused by inadequate network conditions, I introduced packet loss, delay and various bandwidth on the emulation links during my experimentation. Thus, I have created three scenarios with different network conditions in order to evaluate the eMOS of each video sequence. The delay in all scenarios is “1ms”, and the bandwidth is 1Mbps, 5Mbps and 500kbps, respectively in the three scenarios. The Table 6.4.2 presents for each of the two video sequences the different video resolutions, the QPs, the average of rebuffering times (ART), the average duration of the rebuffering in seconds (AD) and the quality switch rate (QSR).

Table 6.4.1: Media parameters for test sequences

Name	Resolution	VB	QP	FR
BBB	360p, 720p	Variable	12, 23, 36	24
ED	360p, 720p	Variable	12, 23, 36	24

Table 6.4.2: Quality Metrics

Resolution	QP	ART	AD	QSR
BBB(H)	12	3	5	2
BBB(H)	23	2	2	2
BBB(H)	36	3	3	1
BBB(L)	12	2	3	2
BBB(L)	23	2	2	3
BBB(L)	36	1	5	1

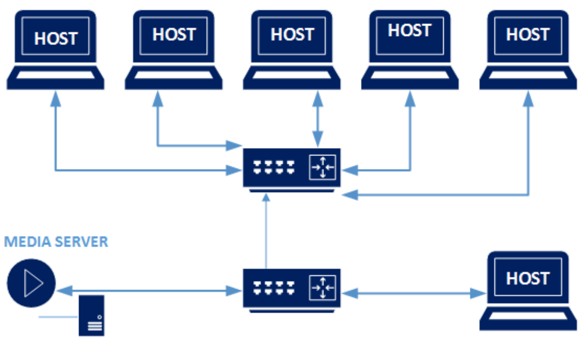


Figure 6.4.1: Network Topology

Table 6.4.3: Quality Metrics

Resolution	QP	ART	AD	QSR
ED(H)	12	3	3	3
ED(H)	23	2	4	2
ED(H)	36	3	4	3
ED(L)	12	2	5	3
ED(L)	23	1	5	2
ED(L)	36	2	2	1

6.4.2 Results

In Figure 6.4.2, I observe that the eMOS is lower in Scenario 1 because, in that scenario, there are more rebuffering events with more duration because of the low bandwidth.

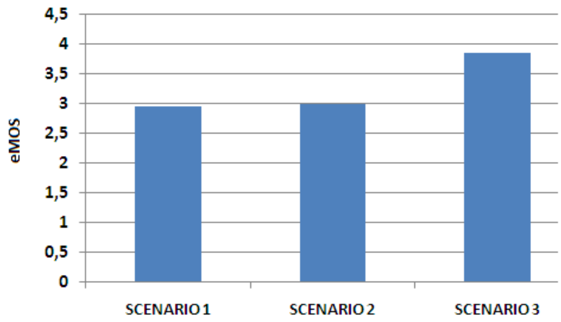


Figure 6.4.2: eMOS results for the 3 scenarios

In Figure 6.4.3, I observe that eMOS is higher than the other cases because the network conditions were better (higher bandwidth).

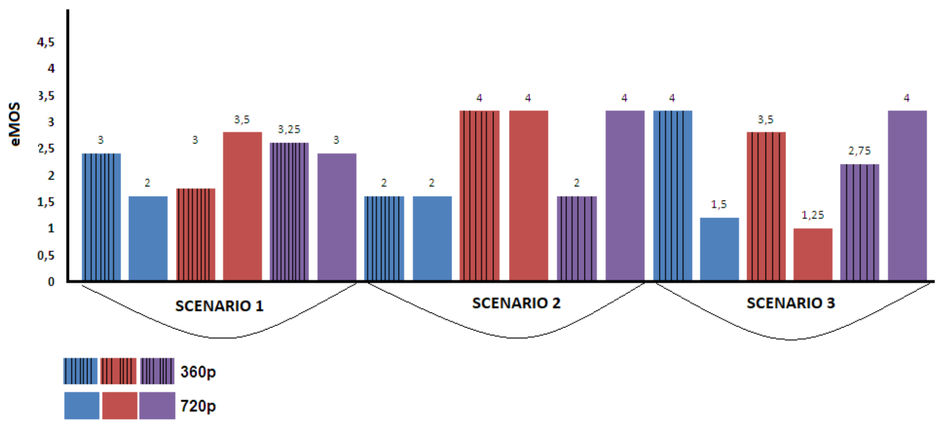


Figure 6.4.3: Evaluated eMOS results considering QP

Moreover, in Figure 6.4.4, I observe that resolution has a significant impact on the eMOS evaluation. The eMOS value is higher when the video resolution is 360p. Note that in both cases the QP is 23 (the official DASH QP value). As a result, I observe higher throughput and lower packet loss rate. Calculating the eMOS, I observe the values of eMOS with different network conditions and different application configurations.

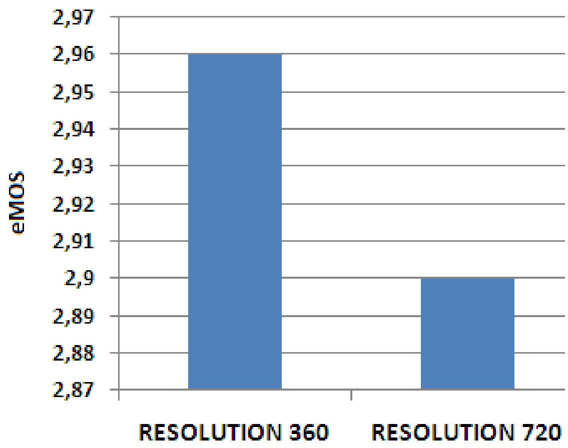


Figure 6.4.4: eMOS results considering video resolution

6.5 Conclusion

In this work, I proposed a strategy that includes a framework and an SDN-based architecture for QoE monitoring and management. First, I analyze the framework and its layers (data acquisition, monitoring and management), and then I apply the components of the framework to an SDN-based architecture. In order to evaluate its performance in a video streaming scenario, I developed a platform which is based on Opendaylight and Mininet. In order to evaluate the quality of the video streaming, I used eMOS model considering rebuffering events, duration of the rebuffering, switch quality rates, video resolution, quantization parameter and network parameters such as bandwidth. The results show that eMOS and its parameters are related to network condition, video resolution and a quantization parameter.

Chapter 7

An Agent-based solution for QoE monitoring

7.1 Introduction

The new generation of Long Term Evolution (LTE) networks provides ubiquitous broadband access to mobile devices matching land communications in quality and speed. However, to optimize network resource usage in a dynamic environment, network operators need models and strategies to constantly assess and manage the end-user's Quality of Experience (QoE). Given the importance of these activities, in the current work, I focus on quality monitoring and the usage of QoE-agents in an LTE-Advanced Pro network. Thus, I have proposed two Agent-based monitoring approaches.

In the first approach, I identify the location and the operation of the QoE-Agents based on the accuracy of the measurements and the load in the network by considering the frequency of the measurements and the running applications. Emulations have also been carried out to evaluate two considered scenarios with different network conditions as well as with different quality sampling rates and application configurations. The preliminary results have shown that the proposed strategy results in acceptable errors from my measurements, low CPU utilization and acceptable memory utilization.

In the second approached, I relied on an approach that considers the computational complexity and the network load to adjust the frequency of measurements while considering the estimation accuracy. Accordingly, the proposed system allows the Internet Service Provider (ISP) to monitor the network QoE level accurately without overloading the network by considering different Influence Factors (IFs). Some preliminary results are shown in terms of accuracy of estimations and computational complexity in case of video streaming service.

7.2 Related Works

In [184], the authors proposed a multi-agent architecture with an embodied end-user autonomous agent that represents personal preferences and expectations and seeks to maximize QoE. Later this approach is evaluated with "Repast", a multi-agent simulation platform. The results show that a decentralized multi-agent QoE-management outperforms a similar centralized approach both in terms of end-user satisfaction and service acceptability. The authors in [185] proposed a framework for collecting QoS-based parameters using a Device-Agent and they focus on the Monitored-Set selection model to collect QoS. Note that, the QoS refers to concepts, and measures of network performance [186] while the QoE defined as "the overall acceptability of an application or service, as perceived subjectively by the end-user" [186]. However, none of the above approaches considered the network load or the computational complexity of implementing these agents.

Recent research had shown that QoE is highly personal and influenced by multiple factors, including the user expectations, preferences and cultural background. In the literature, some existing QoE monitoring and management approaches are centralized. In order to increase the end-user participation in a QoE-based system, I follow a decentralized approach using distributed "QoE-Agents". It is essential for a QoE-monitoring system to be autonomous and to consider the personal preferences of the users to improve his satisfaction [184], [187], [188].

Mobile network operators monitor the QoE in their network by deploying monitoring platforms such as QoE analytics System [189], eMAN [184], QoE-Center [190], or frameworks [188], or architectures such as middleware architecture [187], data-driven architectures [4] or extend existing LTE architecture [191]. The process that they need to follow to monitor and manage QoE includes three main tasks: the data collection, the data manipulation and then applies the appropriate actions to improve the QoE to increase users' satisfaction [189], [191],[187], [190]. Moreover, the monitoring can be done in real-time [189], [191], [187], or in non-real-time [4].

Since QoE is multi-dimensional, the monitoring tools should consider context-related parameters such as the user's location or time of usage, context-related parameters, network-related parameters such as network QoS such as the delay and user-related parameters such as the user's preferences[56]. However, most of the existing approaches are considering some combinations of these parameters, as the following: (i) only context-related and network-related parameters [4], (ii) only user-related and context-related parameters [192] or (iii) only user-related and network-related parameters [188] [193]. Thus, there is a need for a system which considers all the important QoE-related parameters.

The data collection for QoE monitoring can be deployed by an additional entity such as QoE-Center[190], QoE-Orchestrator [187], QoE analytics system [189]. In some cases, the network providers need to install an agent in the users' device/terminal:

- To retrieve user's satisfaction such as the eMAN[184] which aims to represent the user's personal preferences and expectations,
- A mobile agent [4] which collects user's experience using surveys and feedbacks, context-related information using user's location and mobility and network-related information or
- A "QoE-Agent" [194] which is used to show a survey to the user, in order to be aware of the user's experience.

I observe that the main questions/issues are: if I need to follow a centralized or decentralized approach, if I need to install an agent in user's device or not and how I should collect the data. My solution with respect to these approaches is different because I follow a decentralized approach using distributed agents in the network infrastructure to avoid the single point of failure in the network and I prevent the extensive interaction with the user in order to give us feedback, also, in my approach the data collection realized by the agents and not in an additional entity.

7.3 An agent-based QoE monitoring strategy for LTE networks

In this sub-Section, I present my QoE-monitoring solution over LTE-A Pro networks using the QoE-Agents. My target is to apply the QoE-Agents in specific network entities, in order to estimate the QoE level considering the accuracy and the network load. In the following section, I present how I apply the QoE-layered model, which QoE-Agent approach I follow and which models I use to estimate the QoE, the accuracy and the load in the network.

7.3.1 The proposed architecture

The QoE-Layered model

Since QoE-layered model constitutes 6 layers, so someone might think that I need one agent for each layer. However, this is not the case since I employ the PCRF logical entity. For each layer, I retrieve different parameters, execute different processes, define different inputs and outputs from different network components using the PCRF entity that can communicate with the appropriate network entities, through different interfaces. Considering the above, I implement a single "Slave-Agent" in the PCRF entity because the PCRF has a direct connection with other useful entities and one "Master-Agent" in the PGW, which will communicate with the Slave-Agent. I chose to deploy the "Master-Agent" in the PGW because it has direct a connection with the PCRF and with the Internet, e.g. other networks and Over-The-Top (OTT) providers.

The layers of the QoE-Layered model and their related factors are the following:

Resource Layer The factors are related to the characteristics and the performance of the network resources and the technical system which are used to deliver the service. The network resources are related to network QoS (e.g. delay). The system resources are related to the process capabilities of the server and end-users' device capabilities.

Application Layer The configuration factors about the application or the service and factors related to the content are considered, such as frame rate. In this layer, several approaches appear. One possible approach is to collaborate with an OTT content provider [195], such as YouTube so that they can provide us with the needed application-related information. I am also taking into account information about the detected traffic and the session characteristics.

Interface Layer It represents the physical equipment of the user, such as the type of device. Moreover, the interface that the user interacts with the application/service. For this layer, the network provider does not have any knowledge. Thus, other approaches need to be followed. In a user-based approach, the user provides this information, for example, with a series of questions that must be answered, such as a questionnaire. This approach cannot easily be applied because it cannot be used for automation and real-time situations. It is also a time consuming and complicated method [196]. In a probe-based approach, a monitoring probe can be installed to the user's terminal, which extracts this information with respect to a user's privacy [197]. This probe will communicate with the "Master-Agent" in PGW, and this is the optimal choice if the user agreed.

Context Layer It includes all the factors that are related to the physical context (e.g. geographical aspects), the usage context (e.g. mobility/no-mobility) and the economic context (e.g. cost of a service). Firstly, for the physical context, there are some factors that a network provider does not have any knowledge, such as ambient light or noise, and the only way to get any knowledge about these should come from user's side. There are different ways to retrieve this kind of information, as discussed before, and in some cases, even the device cannot export this kind of information. Consequently, the usage context is related to the mobility/no-mobility factor. The information for usage context is known for the network provider through the eNBs. The stress/no-stress factor is difficult to be measured. I must estimate this factor using the psychology of the user and estimate its' psychological situation depending on his actions using the user's device. The economic context is related to the network provider. The PCRF entity of the LTE network includes all the information related to the subscriber. This information is stored in databases, like the SPR that includes information about the allowed services, spending limits and all the related information regarding the subscriber's profile. The information about the economic context is very critical for the QoE assessment and the behaviour of the network[198] (e.g. if a subscriber pays a lot for a service then high quality is expected).

Human Layer It includes psychophysical factors that are related to the perceptual characteristics of users. These factors are nearly impossible to be measured, and they are related to the user as a human being. This means that factors such as the sensitivity to audiovisual stimulus, the personality can be known to the network provider only if the user provides them.

User Layer It is about user's perspective for a service/application and includes factors as level of expertise, patterns, history and social characteristics.

The QoE-Agents

In the proposed solution, I follow the "Distributed" approach and I implement one "Master QoE-Agent" (MA) and one "Slave QoE-Agents" (SA). The MA in the PGW and the SA in the PCRF entity. The MA is responsible for communicating with SA and with external applications, such as QoE-aware application, on the Internet side. The SA is responsible for retrieving data for the following layers and then using them as input for its processes:

- **Resource (L1)**. From the TDF entity, the SA retrieves network parameters in terms of delay, jitter, loss, error rate and throughput.
- **Application (L2)**. From the TDF entity, the SA retrieves information for the detected Application ID and its description. Also, from AF entity, it retrieves application parameters such as the media type, the media format and the name of Application Service Provider.
- **Context (L4)**. The SA retrieves information related to the economic context of the SPR entity.

The MA is responsible for retrieving data for the following layers and then using them as input for its processes:

- **Context (L4)**. The MA retrieves information related to the location of the user from the BBERF entity.
- **Interface (L3)**. The MA retrieve application-related parameters via the collaboration with Application Provider.

Note that, the Interface Layer (L3), the Human Layer (L5) and the User Layer (L6) are related to the user, and I can retrieve information using two approaches. I can install an agent in a user's device so the user can send feedback or to collaborate with an Application provider.

7.3.2 Performance Evaluation

Experimental Setup

In this section, I evaluate the performance of the proposed strategy using the Estimation model from 4.3. I used Mininet network emulation environment [182] in which a media server is used to stream video to the users who are inside Mininet and use VLC player version 3.0. I use network monitoring tools to identify the bandwidth and throughput of the network. Mininet was installed in an Asus computer with Intel Core i7 @3.6 GHz, 4GB RAM installed with Linux Ubuntu 14.04, 64bit.

The experiment is carried out using the experimental setup shown in Figure 7.4.1. The network includes five hosts, one media server (in the Application Provider) and 11 network devices which are considered as the logical entities of an LTE-A Pro network. The Master-Agent is installed in the PGW, and the Slave-Agent is installed PCRF.

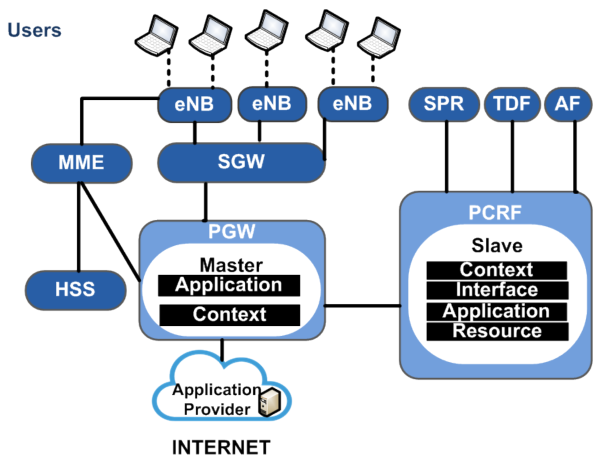


Figure 7.3.1: Agent-based QoE monitoring in LTE-Advanced Pro network

I have created two scenarios with different network parameters such as delay, bandwidth, jitter, packet loss, and application parameters such as video resolution to evaluate the performance of my strategy. My target is to observe if the estimated QoE level in a video streaming session is affected by the network conditions and by the frequency of the measurements as well as the video resolution has an impact on that. A specific QoS level is provided, and I used Eq.4.18 to estimate that level. I introduced loss, delay and various bandwidth on the emulation links during my experimentation. I evaluate the Estimated QoE of each video sequence. The average end-to-end transmission delay for each path in the network in all scenarios is 4ms. The available bandwidth fluctuates between 5 Mbps and 10 Mbps, respectively in the two scenarios. In each scenario, the media server serves at least two clients, and I have added additional background traffic in order to have real network conditions. Background traffic is another type of traffic which is used simultaneously along with the primary traffic of interest (in my case the traffic came from video streaming sessions). Under ideal network conditions, I have all the necessary network resources and bandwidth so I cannot prove that my strategy for a video streaming service will work better even in the worst network conditions. Thus, for that reason, I need to simulate the worst-case using bottlenecks and traffic overhead. Furthermore, in order to find the ideal frequency of the measurements, I have executed the four processes every 1 minute, 3 minutes, 5 minutes, 7 minutes and each second for 30 seconds. My target was to observe which is the ideal frequency of measurements, in order to avoid overloading the network and consuming many resources in terms of CPU and memory and keeping the QoE at an acceptable level.

I have developed the following processes which are executed by the agents in the emulation environment with this order:

1. "QoS level" process is related to Eq.4.18 and calculates the QoS level for the best path [199] from the user to the App. Provider considering the bandwidth, the delay, the jitter and the packet loss. The SA executes this process, and it is related to "Resource Layer" of the QoE-layered model.

2. "Est. QoE" process is related to Eq.4.19 and estimates the QoE level for the best path [199] from the user to the Application Provider considering the "QoS level", the resolution of the video, the GOP and the video quality. The MA executes this process, and it is related to "Application Layer" of the QoE-layered model.

3. "Utilization" process measures the CPU and Memory utilization when the agents are used to monitor the network. It is executed in the background in my system, in order to observe if the utilization if decreased or increased.

In order to measure the exact values of bandwidth, delay, jitter and loss, I used network analyzing tools such as iperf [200] and qperf[201]. I also used "Big Buck Bunny" (BBB)[99] video sequences, considering four different resolutions such as 240p, 360p, 480p and 720p and the H.264 codec. Moreover, the frame rate was 24fps, and the container was mp4. In Figure 7.3.2 is shown the block diagram of the proposed monitoring strategy.

Results

In this subsection, I describe the experimental results based on the two scenarios with different parameters, also considering the video resolution. Note that the values in the plots are normalized in order to be able to compare the values. All the QoS values are normalized to be between [0-5]. Consequently, in each scenario, I have the following frequencies of measurements: 1 minute, 3 minutes, 5 minutes, 7 minutes and a 30seconds continuing observation. For these, I estimate the QoS, QoE, error (using MSE), the CPU utilization and memory utilization when the video resolution is 240p, 360p, 480p and 720p.

In Figure 7.3.3 and Figure 7.3.4, I did measurements in each second during the video session about the QoS and QoE levels, respectively. Figures 7.3.3 and 7.3.4 show that in Scenario 1, the QoS and QoE levels during a session have higher variation due to the network conditions because there are more delays and higher jitter. However, the results in Scenario 2 seems to be more stable. Thus, I understand that the better network conditions of Scenario 2 help the QoE level to be higher and more stable.

Figure 7.3.5 illustrates the average estimated QoE values, whereas Figure 7.3.6 shows the average QoS values during a session for both Scenarios. For Scenario 1 in Figure 7.3.6, I observe that the network parameters are better for 240p because the load is lower than in the other cases. In order to prove that the frequency of the measurements has a significant impact on QoE level, I include in Figure 7.3.7 also the estimated QoE values when the sampling is happening every 1 minute, 3 minutes, 5 minutes and 7 minutes. Thus, I observe that the better network conditions in Scenario 2 and the different sampling rate has a significant impact on QoE level.

In order to estimate the accuracy of measurements, I used the MSE metric. Results in Table 7.3.1 show that I achieve higher accuracy in Scenario 2. The memory utilization seems to be extremely high; however, this is due to the capabilities of the virtual machine that I used. The CPU utilization is low and has a stable increasing.

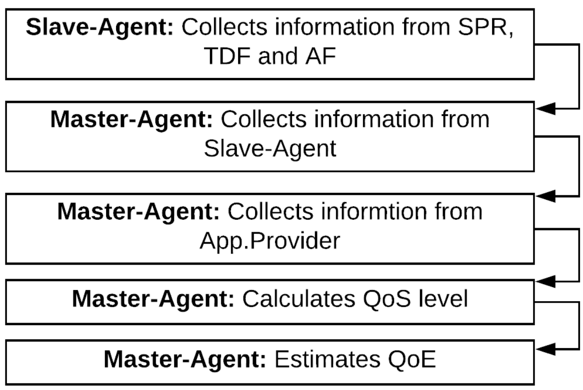


Figure 7.3.2: The QoE monitoring process

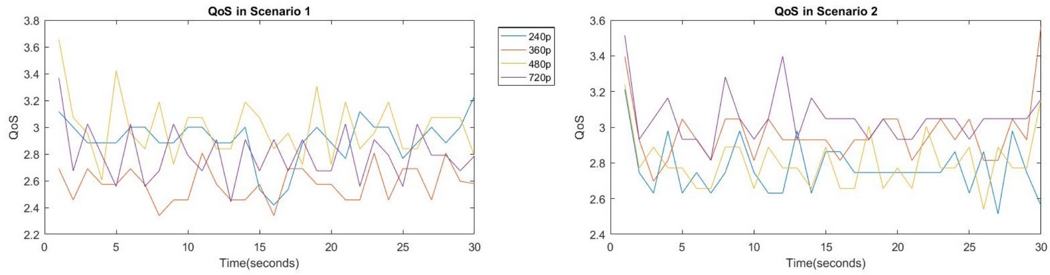


Figure 7.3.3: QoS value for each scenario during a video session

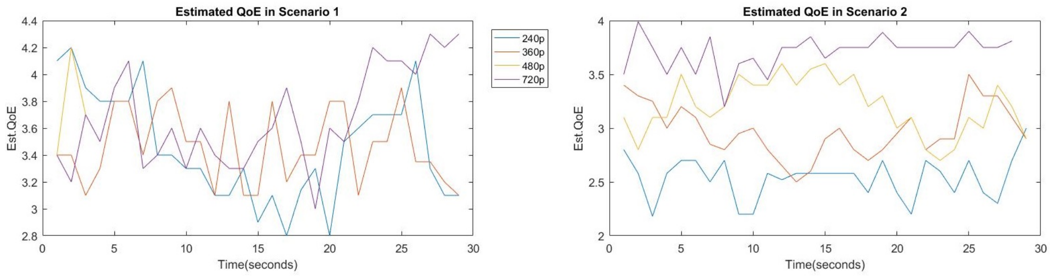
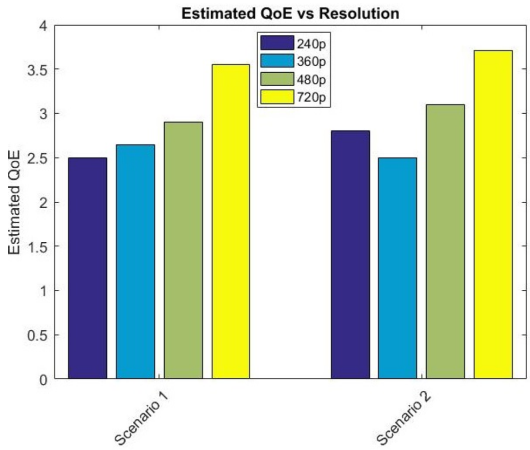


Figure 7.3.4: Estimated QoE for each scenario during a video session

7.3.3 Conclusion

QoE monitoring of video services over wireless networks is essential for performance optimization. In this paper, I developed a QoE monitoring strategy based on a QoE-Layered model and QoE-Agents. Considering the QoE-Layered model, the QoE-Agents have been implemented in two logical entities of an LTE-Advanced Pro network, in order to monitor QoE level during a video streaming application. The QoS parameters were adjustable to meet the real network conditions. In collaboration with the Application Provider, I retrieve application parameters. The QoS parameters, the application parameters, the CPU utilization and memory utilization, are considered, in order to estimate which is the ideal frequency of measurements in an LTE-Advanced Pro network. I have considered two scenarios with different network conditions and I did experiments with different sampling rates and video resolutions. From the preliminary results, I observe that the sampling rate, the video resolution and the network parameters are important for the QoS measurement and QoE estimation. Furthermore, I have acceptable errors from my measurements, low CPU utilization and acceptable memory utilization. My future work includes utilizing a different QoE estimation model as well as different application scenarios such as VoIP in order to achieve higher accuracy and observe the behaviour of the strategy with different network traffic.



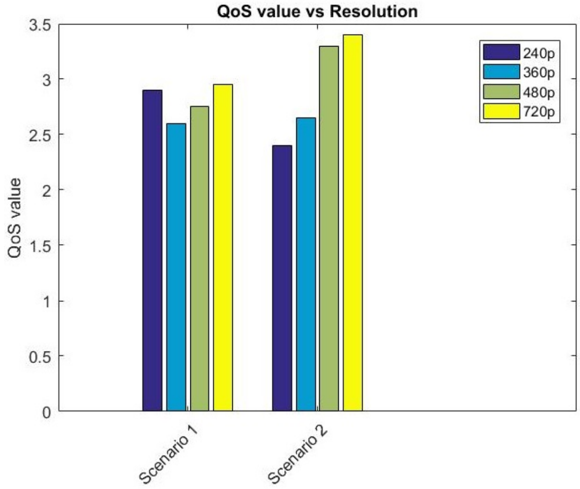


Figure 7.3.5: Estimated QoE value for each scenario

Figure 7.3.6: QoS value for each Scenario

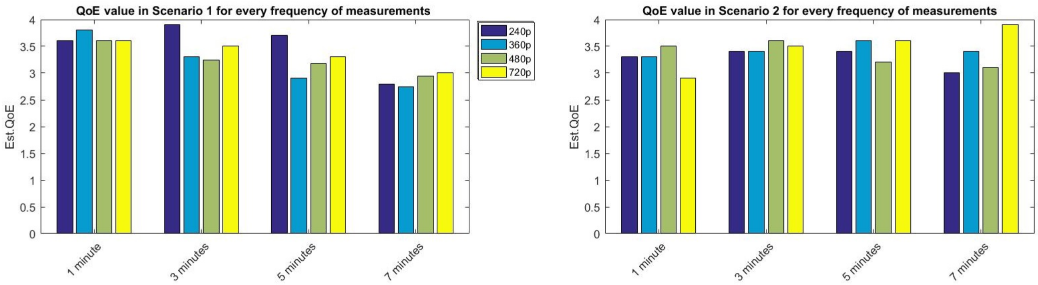


Figure 7.3.7: Estimated QoE value for every frequency of measurements

7.4 QoE monitoring using an agent-based solution

This Section presents my QoE monitoring solution that relies on the use of “QoE Agents” in specific network entities of the LTE-Advanced Pro network to estimate the QoE level while considering the running applications, network load and required computational resources. design has been driven by the following questions: (1) Where the QoE agents should be installed? (2) Which are the IFs that I should consider to estimate QoE level? Furthermore, (3) how should the QoE-monitoring procedure work?

7.4.1 The proposed architecture

The QoE-Agents

The collected data belong to multiple dimensions; thus, it is more efficient to use distributed agents to collect them. It is a challenge to consider a variety of IFs because I need to deal with a diversity of collected data and to decide which parameters are needed to estimate the QoE of service in a mobile context. One benefit of the proposed architecture lies in the fact that the “QoE Agents” are distributed, and they can retrieve

Table 7.3.1: Accuracy estimation with MSE, CPU and Memory utilization during a video session

Scenario - Resolution	MSE	CPU Util.	Mem. Util.
1 - 240p	0.65	15, 11%	95, 49%
1 - 360p	3.61	12, 47%	95, 93%
1 - 480p	1.8	24, 9%	96, 59%
1 - 720p	1.31	26, 76%	96, 89%
2 - 240p	6.48	16, 9%	94, 26%
2 - 360p	6.74	11, 96%	96, 98%
2 - 480p	5.5	18, 94%	96, 86%
2 - 720p	6.3	23, 76%	96, 80%

information related to QoE IFs such as application, context and network IFs. I propose a "QoE Agent", that includes 3 agent types; one "Master QoE-Agent" (MQA) and two "Slave QoE Agents" (SQAs) as shown in Figure 7.4.1.

- The "MQA" is deployed in the PGW because it can communicate with the rest of the Internet and acquires data from the SQAs and forwards it to the Cloud platform and to the internal system to store and analyze them using the standard network management protocols. The ISP can choose to store/analyze the data in an external Cloud platform or his system locally. Also, it can retrieve application-related parameters ("Application" IFs) and context-related parameters ("Context" IFs) if there is a collaboration between the ISP and the Application Provider[202]. The ISP does not have direct access to application parameters by inspecting only the network traffic because the last years the Application Providers encrypt their data. Thus, if there is a collaboration between them, the ISP can retrieve application parameters from the application provider. Considering this collaboration, the Application Provider can monitor application parameters and context parameters through the application software, and forward this information to the ISP through a standard interface. However, this collaboration is out of the scope of this paper. Based on the service type, MQA can use different QoE estimation models.

- The "SQA 1" is deployed in the SGW to retrieve information from the E-UTRAN. It can retrieve information about the usage context ("Context" IFs) concerning mobility through the eNBs. I envision a collaborative network of personal devices as depicted in the SIoT [60] which collaborate for sharing information among them assuring privacy and that the object share only owner-authorised data. Thus, context-related and interface-related ("Context" IFs) information can be retrieved from a SIoT network using the relationships between the SIoT objects. By utilizing the SIoT concept, I can know the type of user's device, the model or the location of the device. Location reporting can be used to support a variety of applications. SGW includes "User location information (ULI)" [203] feature that includes location-related data, such as Tracking Area Identity, Mobile Country Code, Mobile network code and Tracking area code. Thus, ULI feature is another option to retrieve a user's location.

- "SQA 2" is deployed in the PCRF to retrieve the network-related parameters such as throughput through the TDF entity ("Network" IFs), the application-related parameters such as Application ID and its description through the TDF entity ("Application" IFs), and the economic-related parameters through the SPR entity ("Context" IFs). Furthermore, PCRF includes information about the usage context of the service ("Context" IFs) such as the user's location and time zone. [31].

Note that, user-related parameters can be retrieved by the installation of an agent in the user's device so the user can send feedback to the ISP. Thus, the answer to the question (1) is that the "QoE Agents" are installed in 3 network components, the PGW (MQA), SGW (SQA1) and PCRF (SQA2). Note that, considering the size of the network multiple "QoE Agents" can be deployed. The answer in question (2) is that the MQA can retrieve application-related and context-related IFs, the SQA1 can retrieve context-related IFs and the SQA2 can retrieve context-related, network-related and application-related IFs. The proposed QoE-aware architecture using QoE agents and their QoE-related IFs is shown in Figure 7.4.1.

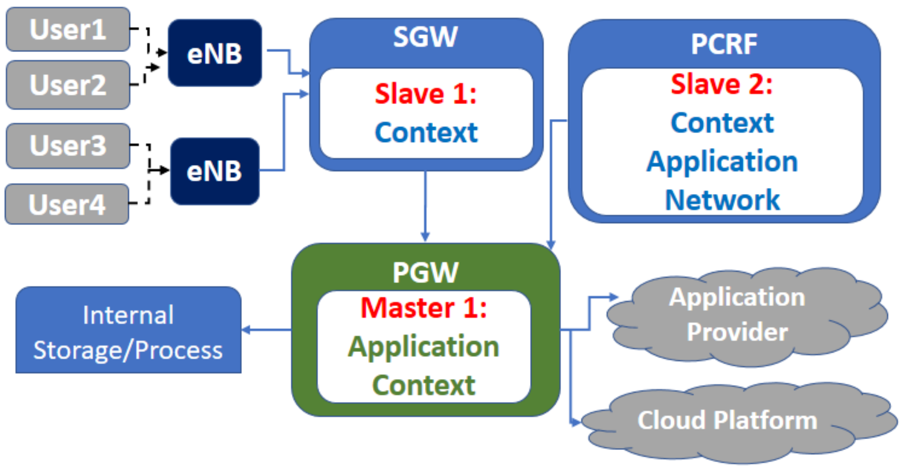


Figure 7.4.1: QoE-aware architecture using QoE Agents and the QoE-related Influence Factors

7.4.2 The monitoring algorithm

To answer the question (3), I present the workflow of the proposed monitoring algorithm in Figure 7.4.2. This algorithm includes 2 phases, the "Monitoring Phase" (orange boxes, steps 1-7) and the "Report Phase" (blue boxes, steps 8-10). The "Monitoring Phase" includes three tasks which are (i) data collection, (ii) KPIs extraction and (iii) QoE estimation. The "Report Phase" includes the reporting of the monitoring. The steps of the algorithm are the following:

1. To start the QoE monitoring process, the network manager defines the QoE monitoring type through the "Monitoring Application Interface". The monitoring type can be one of the following, (i) location-based (e.g. monitor specific part of the network), (ii) user-based (e.g. monitor a specific user) or (iii) service-based (e.g. monitor a specific service). The time scale is defined in "Step 1", e.g. if the QoE monitoring is on-per session basis.

2. The "QoE Agents" are initially configured to make measurements every 3 minutes [204].

3. The SQA1 retrieves context-related information such as user's mobility from the eNBs, and if there is a SIoT network, it retrieves data for the user's device and its location through the relationships of the objects (devices). The SQA2 collects network, context, and application-related parameters using standard APIs. Context-related information such as location and timezone included in the PCRF; thus, the SQA2 already knows them, and it can use them. Also, context-related information such as economic parameters retrieved through the SPR entity. Through TDF entity, it extracts network-related parameters such as throughput, delay, bandwidth and application-related parameters such as application description and application type.

4. Both SQAs forward their data to the MQA. MQA has a direct connection with the Internet and can forward the data to the Cloud platform using the REST protocol and to the local storage for storage and analysis. Note that, any Cloud platform can be chosen by the ISP, if it supports REST APIs, such as ElasticSearch [205].

5. When the Cloud receives the data, it identifies the application type, extracts the related KPIs such as latency, QoS per application, utilization, network traffic using data mining techniques such as classification [206]. Then, it forwards the KPIs to the MQA.

6. Consequently, the MQA includes the QoE models, to estimate the QoE based on the application type and the identified KPIs. Considering the service type, different QoE estimation models can be used. Note that, the KPIs are related to accessibility such as traffic flow, retainability such as throughput and integrity such as data streaming quality, throughput, delay, jitter [207] so that KPIs may differ from service to service.

7. The results from the QoE estimation are stored in the internal database and to the Cloud storage.

8. During the monitoring and estimation procedure, the "Monitoring Report" function measures the network load, the network latency and the computational resources regarding CPU and memory utilization during the QoE monitoring. The results from the "Monitoring Report" function are saved in JSON format in the local storage and forwarded to a cloud analytics system via a REST API. The network administrator can observe and analyze these results by creating his dashboards and then apply the proper management technique to improve the accuracy of the QoE estimations [208]. Based on these measurements, the frequency of measurements is adjusted to enhance the accuracy of the estimations.

9. When the QoE estimation is completed for the first run, the algorithm checks if the network load during the monitoring procedure was high and then it adjusts (increases) the frequency of measurements properly by repeating the monitoring procedure (Step 2).

10. If the network load was not high, then it checks if the computational resources were high. If they were higher during the monitoring procedure, then it adjusts (increases) the frequency of measurements properly by repeating the monitoring procedure, (Step 2) to decrease them.

Thus, the proposed algorithm aims to adjust the frequency of measurements to increase the accuracy of the QoE estimations while at the same time keeps the computational resources and the network load at a low level. Thus, based on the data collected from the SQAs and the extracted KPIs the network administrator will have better insights about the user's experience considering QoE IF such as the user's location, timezone, mobility, service type, network conditions and service cost.

7.4.3 Performance evaluation

Experimental Setup

In this section, I evaluate the performance of the proposed solution. The Mininet emulation environment and network monitoring/analyzing tools have been used to emulate the network. With Mininet, I can identify the network load and the latency, and estimate the QoE level. Mininet was the most appropriate emulation environment because it is a favourite emulation tool among the researchers. As a big data platform, ElasticSearch, Logstash and Kibana (ELK) stack were used [205]. With ELK, the network administrator can analyze data and create a dashboard. The ElasticSearch is a database with search capabilities which uses JSON. Logstash is used as a pipeline to

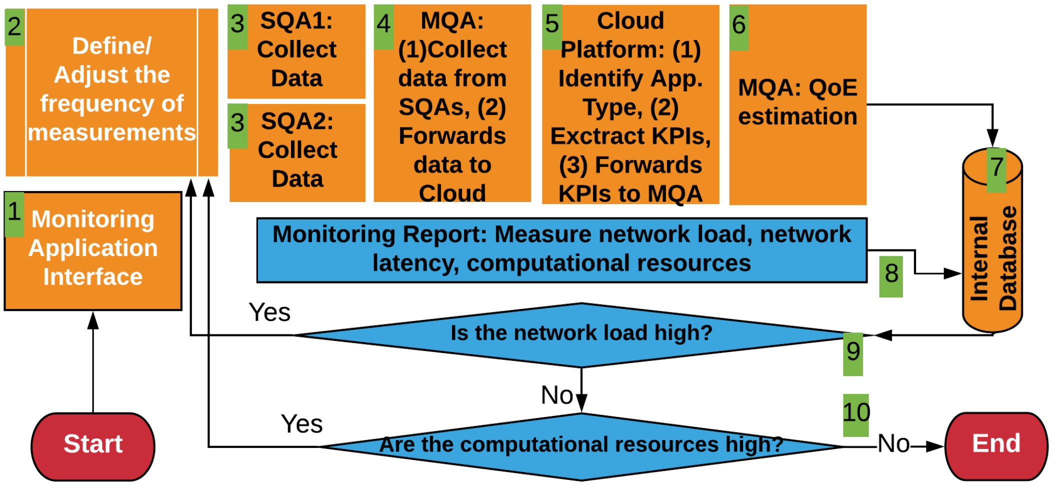


Figure 7.4.2: Workflow of the proposed agent-based QoE monitoring solution

retrieve data from anywhere. It is called Extract, Transform, Load (ETL) pipeline and allows the fetching, transforming and storing events into ElasticSearch. The Kibana is a web-based data analysis and dashboarding tool for ElasticSearch. It visualizes data in seconds from ElasticSearch.

The network topology considered in this analysis includes four hosts (users) and five devices considered as the logical entities of an LTE-Advanced Pro network, as shown in Figure 7.4.1. The logical entities are 1 Service Gateway (SGW), 1 Packet Gateway (PGW), 1 Policy Control and Rules Function (PCRF) and two eNodeBs (eNBs). I assumed that the links from the users to the eNBS were wireless. The MQA was deployed in the PGW and the two SQAs to the SGW and PCRF.

I have created two scenarios. In the first scenario, the sampling frequency during a video streaming is constant every 3 minutes as proposed in [204], whereas in the second scenario I used the proposed algorithm to adjust the sampling frequency following the workflow of Figure 7.4.2. The monitoring frequencies that have been tested were of one sampling every 1, 2, 3, 4, 5, 8 and 10 minutes. The encrypted traffic has not been considered as the case of encrypted traffic is out of the scope of this paper. The average transmission delay for each path in the network was 3 ms, and the available bandwidth was 5 Mbps [204]. The target was to analyze the variation in the accuracy of the estimated QoE by adjusting only the monitoring frequency. Using the proposed QoE agent approach, the aim is to avoid network overload, to consume fewer resources regarding CPU and memory as well as to keep the QoE at an acceptable level by changing the monitoring frequency.

In the first scenario, without the proposed QoE monitoring algorithm, the monitoring frequency was stable. However, in the second scenario, there was a dynamic adjustment of the frequency.

Table 7.4.1: QoE estimations accuracy using MSE during a video streaming

Time between consecutive samples (min.)	**Without the proposed algorithm**	Time between consecutive samples (min.)	**With the proposed algorithm**
3	0.079	3	0.067
3	0.072	2	0.070
3	0.076	5	0.060
3	0.079	4	0.070
3	0.074	8	0.075
3	0.076	3	0.069
3	0.080	2	0.075
3	0.073	3	0.071
3	0.080		
3	0.089		

Results

The results from the ELK are shown in Figures 7.4.3 and 7.4.4. The network manager can extract results such as average values of estimated QoE, CPU utilization, throughput, jitter or loss, and create graphs such as in Figure 7.4.3 after the data analysis. The QoE level is not low; it is between [0-1] as proposed by the QoE, estimation model.

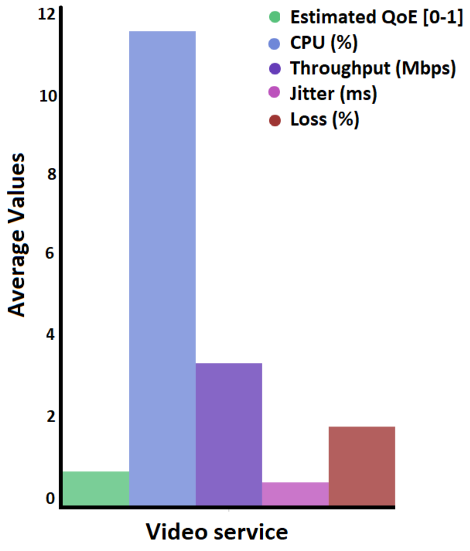


Figure 7.4.3: Video service results with the algorithm using ELK stack

Moreover, the network manager can observe how many services were realized and their type during the monitoring session in a specific monitoring network area. In my case, it exists only a video streaming service and a web browsing service, as shown in Figure 7.4.4.

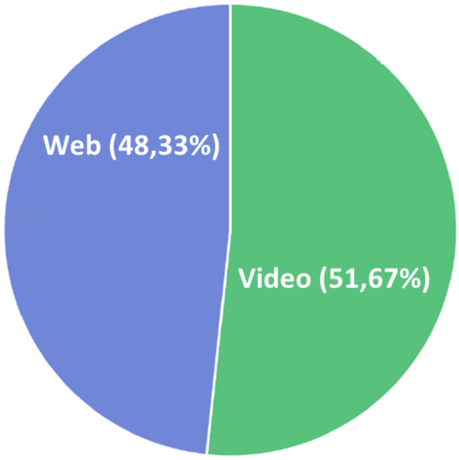


Figure 7.4.4: Identified services during a monitoring session

7.5 Performance Evaluation - Extended results

In this section, I based on my past works [209] [9] and I conducted further experiments in order to meet my objectives, which are the following:

1. Observe the computational complexity in terms of CPU and Memory utilization, when "Approach 1" or "Approach 2" are used under different network parameters and different network topologies.

2. Observe the impact of video containers in terms of computational complexity.

3. I identify an analogy between Agents (Master and Slaves) and the number of hosts in order to keep the computational complexity in low levels.

7.5.1 Experimental Setup

All the experiments were conducted in the Mininet [210] emulation environment. It was selected because it can configure many parameters of a communication channel such as jitter, throughput, delay. To accommodate a large number of networking instances such as switches and hosts, it uses process-based virtualization in a single OS kernel. My Mininet emulation environment was in the Linux Ubuntu server 14.04 (64bits) in Virtualbox environment with 4GB RAM, 10BG Hard drive and Core 2 Duo @2.4GH CPU resources. The two videos that I used in my experiments were the Big Buck Bunny and an extreme sports video 60 seconds long video sequence each one. The first video was considered as the "low motion" video and the second as the "high motion" video. The media server was running as a service in one of the Mininet hosts.

In my past two works [209] and [9], I had proposed two approaches related to the QoE Agents. In the first approach [209], I have proposed one QoE Agent that composed by a Master Agent and a Slave Agent. In my second approach [9], I have proposed one QoE Agent that was composed by a Master Agent and two Slave Agents.

The QoE evaluation is influenced by different parameters such as network and application. Thus, in these extended results, I am taking into account different network parameters (e.g. delay, bandwidth, packet loss and jitter) and application parameters (e.g. video type, resolution and container type). In the simulation setup, in order to achieve my targets, I applied the following: I used two different network topologies. One with 30 hosts as shown in Figure 7.5.2 for “Approach 1” and another with 30 hosts for “Approach 2” as shown in Figure 7.5.3. The second topology includes 60 hosts for “Approach 1” as shown in Figure 7.5.4 and another topology with 60 hosts for “Approach 2”. I used different network topologies in order to find the analogy between Agents and hosts. The applications parameters that I considered were the following, (i) the video type, e.g. low motion and high motion, (ii) the resolution of the video, e.g. 480p and 720p, and (iii) the type of the container, e.g. MPEG-4 and H.264. The network parameters that I considered were the bandwidth, the packet loss, the delay and the jitter. Thus, I created three different scenarios with different network parameters. The "Scenario 1" is characterized as "Great" because it has the ideal network conditions. The "Scenarios 2" is characterized as "Good" and the last scenario, "Scenario 3" is the scenario with the worst network conditions. The values of each network parameters are shown in Figure 7.5.1. Furthermore, in order to estimate the QoE level, I used two QoE estimations models, the eMOS model which is based on application and network related parameters [98] and the PSNR which is the ratio of the peak signal expressed in dB. I measured PSNR, and then I converted to MOS value [211]. In my experiments, I used each approach in each topology, under the different application and network parameters and using the different QoE estimation models.

The main differences between these experiments and my previous experiments are the following:

1. In all scenarios, the videos sequences were played 30 times (30 runs).

2. I compare the results for both approaches in order to find the best case for each one.

3. The network topologies are more and bigger. In these experiments, I have two network topologies. One with 30 hosts and one with 60 hosts.

4. In these experiments, I used 2 QoE estimation models in order to observe if the computational complexity is increased when the QoE model is different. The selected QoE models retrieve different parameters in order to estimate the QoE level of video streaming.

5. The quality sampling rate is dynamic, and the agent-based QoE monitoring algorithm from [9] is used.

6. In these experiments, I consider two application parameters, the video resolution and the video type.

7.5.2 Results

How the network conditions impact the computational complexity in terms of CPU and memory utilization of the VMs in which the QoE Agents were installed?

Observations for Figures 7.5.6 and Figure 7.5.7:

- When the network conditions are "Great" and the video resolution is 480p, the number of hosts and the type of video (low or high motion) has not any impact on CPU utilization because these network conditions can support any video type and any number of hosts without any problem.

- When the network conditions are "Good", the video resolution is 480p, and the topology has 30 hosts, the video type has an impact on CPU utilization because the high motion video (Video2) needs more network resources in order to achieve a good QoE level.

- When the network conditions are "Great" to "Good" and the video resolution is 480p, only with Approach 1 (2 MQAs, 2SQAs), the CPU utilization has a slight decrease. In all the other cases the increment is more than 10%. The network load is not increased due to the low motion video; thus, the CPU utilization is not increased. Moreover, with low motion video and with any approach and number of hosts, memory utilization is not influenced. The network load is low due to the low resolution; thus, it is easier the streaming of this video under these conditions.

- When the network conditions are "Good" to "Bad", the video resolution is 480, and the video type is low motion, I observe that in case of the topology with the 60 hosts, the CPU utilization is not influenced a lot. The network conditions are already bad, and the increased network load cannot affect more CPU utilization.

- For every network condition (Great, Good, Bad), when the video has high motion, and its resolution is 480p or 720p, I observe high memory utilization because the high motion video demands more network resources in order to achieve an acceptable QoE level.

How a video container and the video resolution impact the computational complexity in terms of CPU and memory utilization?

Observations for Figures 7.5.8 and 7.5.9:

- When the network conditions are "Great", the topology has 30 hosts, with any video with resolution 720p and "avi" as container, I observed that the approach and the video type does not influence the CPU utilization because these network conditions can support any video type (high or low motion), video with high resolution and an increased number of QoE Agents.
- When the network conditions are "Great" with Approach1 or "Bad" with Approach2, the video has low motion, its resolution is 720p, and its container is "avi", I observe similar behaviour. The approach, in combination with the container, influence the CPU utilization. More specifically, the network load (and thus the CPU utilization) is increased due to the fact that the "avi" video is heavier.
- When the network conditions are "Bad" with Approach1 or "Good" with Approach2, in a topology of 60 hosts, the video resolution is 720p, and its container is "avi", I observe that the video type influences CPU utilization because when its high motion with high resolution, more resources are demanded in order to achieve an acceptable QoE level.

Observations for Figures 7.5.10 and 7.5.11:

- With Approach 1, when the network conditions are "Great" in the topology of 30 hosts, with any video of any resolution with any container, I observe that the video type and the container do not affect the Memory Utilization and it has the lowest values.
- When the network conditions are "Bad" in any topology, with any video type of any resolution with any container, I observe that the approach and the number of hosts have an impact on the Memory utilization since when the number of hosts is increased, there is a need for more resources in terms of memory.
- When the network conditions are "Great" or "Good" in the topology of 60hosts, with any video type of any resolution with any container, I observe that the approach does not influence the Memory Utilization. The network load is low, and the system can monitor and estimate the QoE level without problems and without increase the Memory Utilization.

How the type, the resolution and the container of a video affect the QoE estimation?

Observations for Figure 7.5.12, Figure 7.5.13, Figure 7.5.14 and Figure 7.5.15:

- In most of the cases, the values of the PSNR to MOS are higher than eMOS model. This is related to the models.
- eMOS also considers the network conditions; that's why its value is lower when the network conditions are not optimal.
- In case of video2 (high motion) with the "mp4" container, the eMOS and the PSNR values are higher than video1 (low motion). This is happening even under "Bad" network conditions. Thus, I observe that the video type, in this case, does not have big influence under these conditions, e.g. video resolution is 720 and any network condition.

7.6 Conclusion

This paper proposed a monitoring solution that uses QoE agents and a QoE monitoring algorithm. My solution first defines the functionalities of the components (QoE agents) involved in a QoE monitoring process by considering a variety of QoE IFs. Then, I collect, analyze and store data related to IFs to estimate the QoE of a video service. My algorithm considers computing resources and network load to adjust the frequency of measurements targeting to make more accurate QoE estimations. I realized that the dynamic adjustment of monitoring frequency has a significant impact on the accuracy of the QoE estimations.

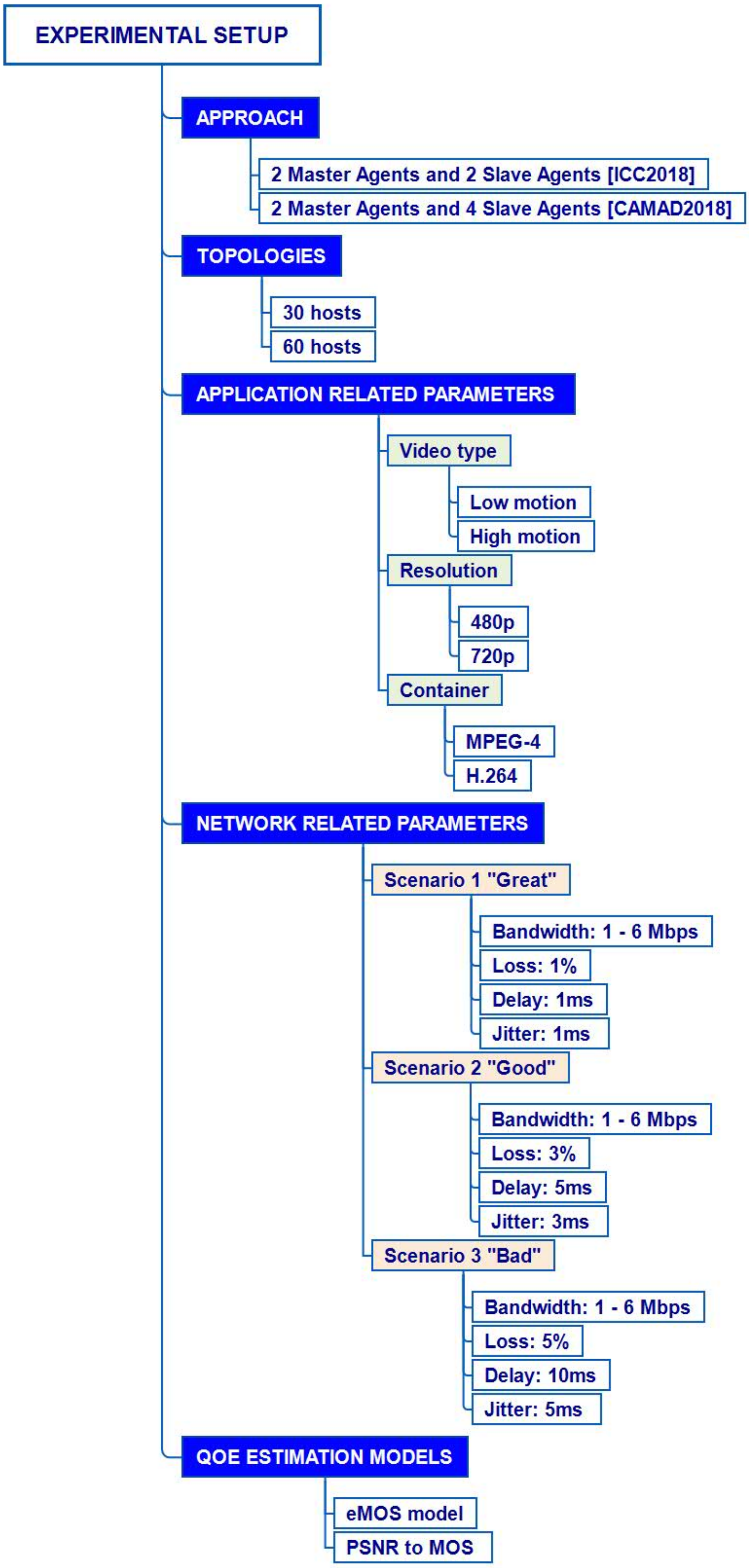


Figure 7.5.1: Experimental Setup parameters

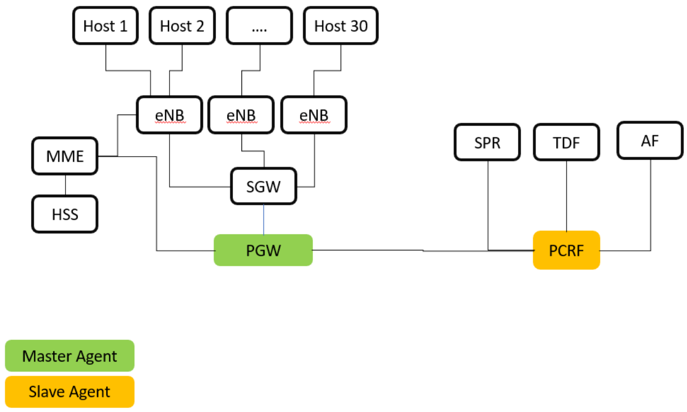


Figure 7.5.2: “Approach 1” with 30 hosts

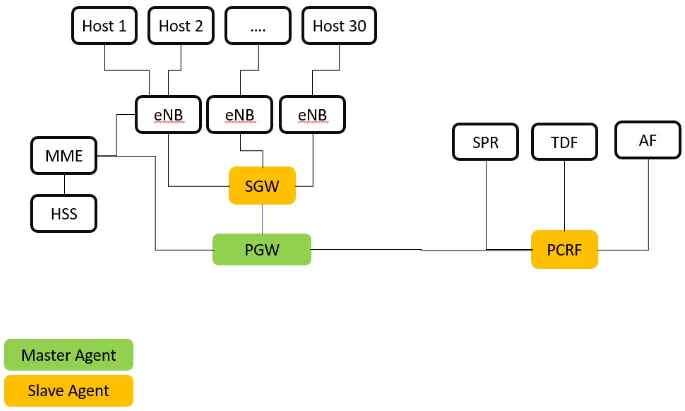


Figure 7.5.3: “Approach 2” with 30 hosts

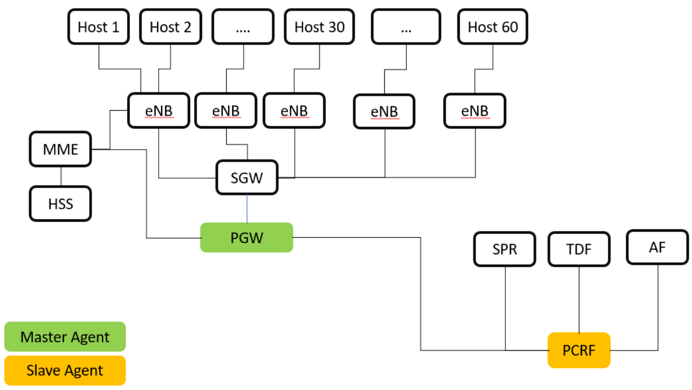


Figure 7.5.4: “Approach 1” with 60 hosts

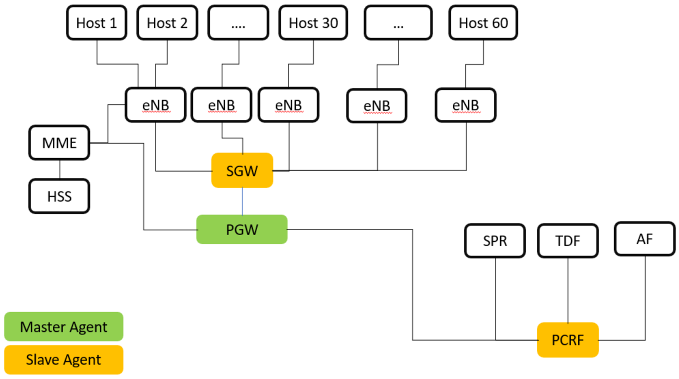


Figure 7.5.5: “Approach 2” with 60 hosts

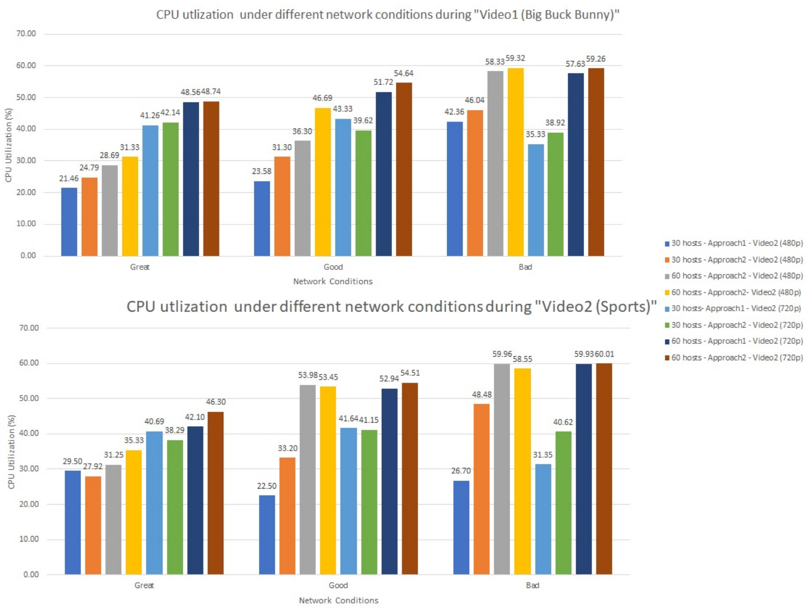


Figure 7.5.6: Network conditions vs CPU Utilization

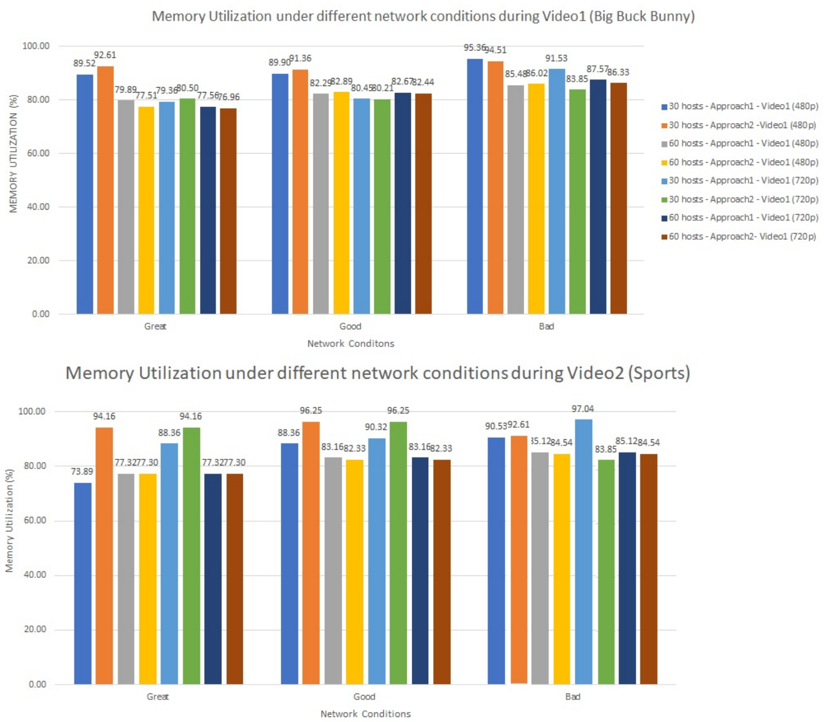


Figure 7.5.7: Network conditions vs Memory Utilization

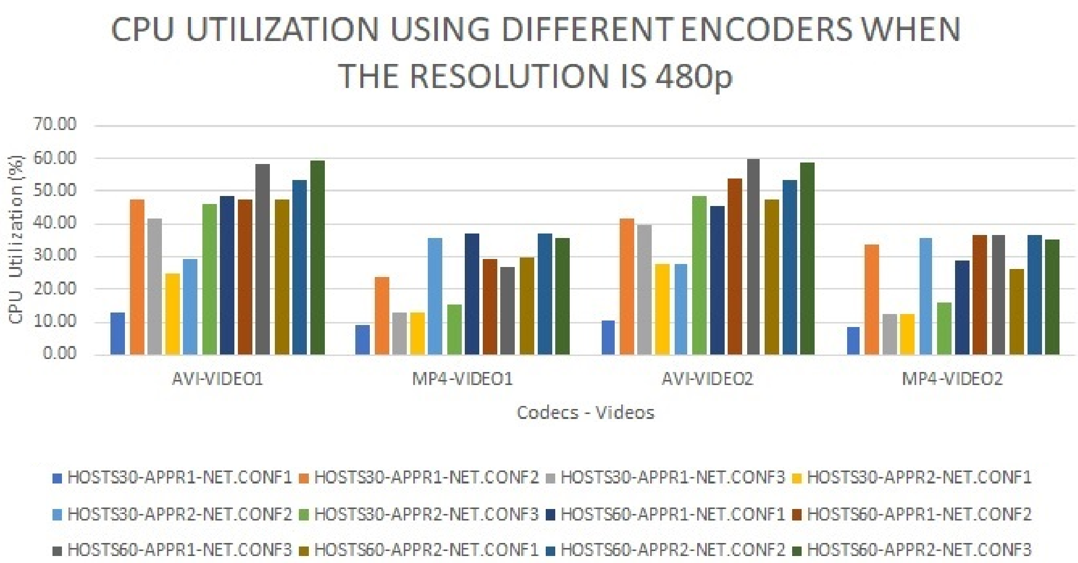


Figure 7.5.8: Computational complexity in terms of CPU Utilization using different containers when the resolution is 480p

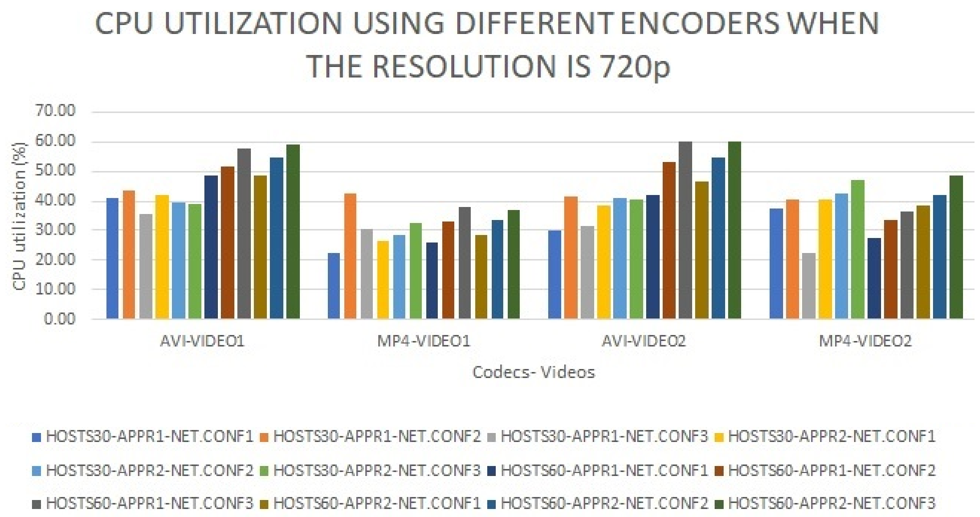


Figure 7.5.9: Computational complexity in terms of CPU Utilization using different containers when the resolution is 720p

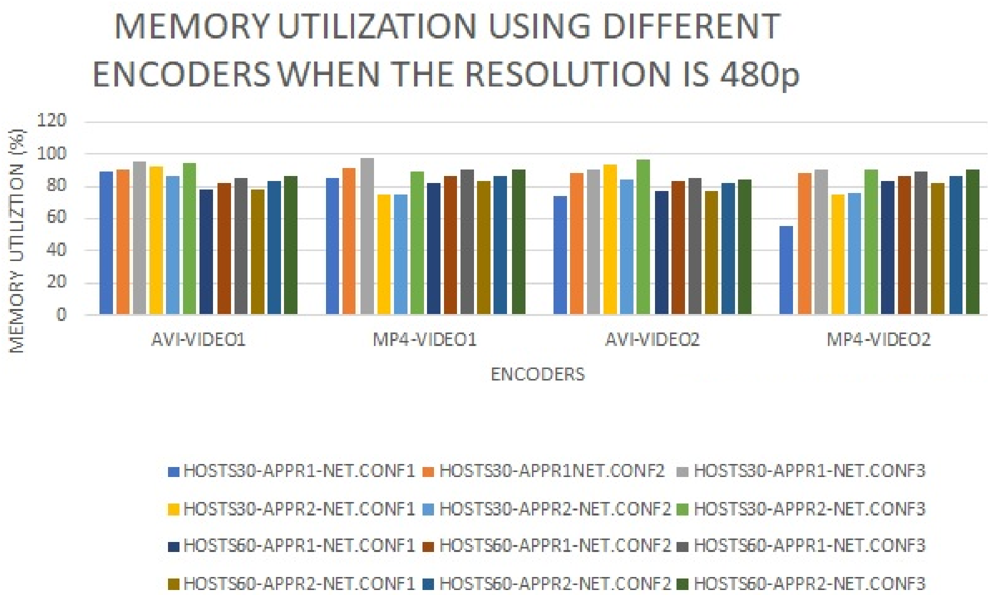


Figure 7.5.10: Computational complexity in terms of memory utilization using different containers when the resolution is 480p

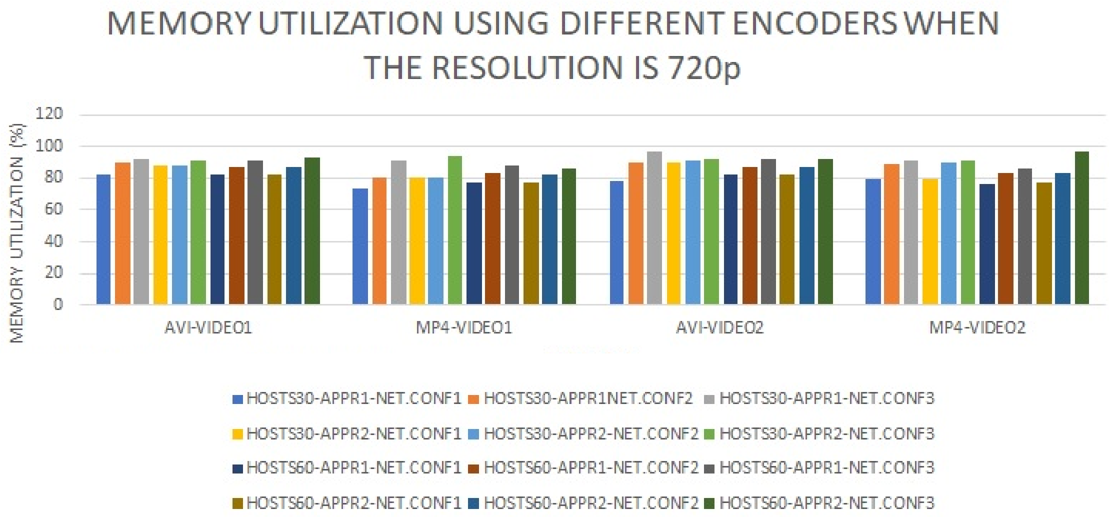


Figure 7.5.11: Computational complexity in terms of memory utilization using different containers when the resolution is 720p

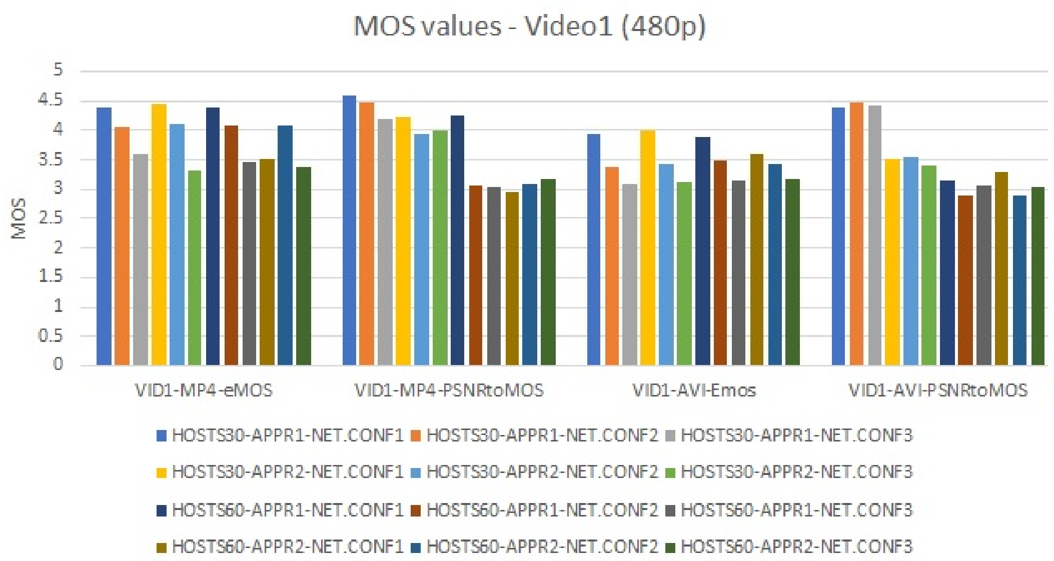


Figure 7.5.12: Average MOS estimation for video 1 when the resolution is 480p.

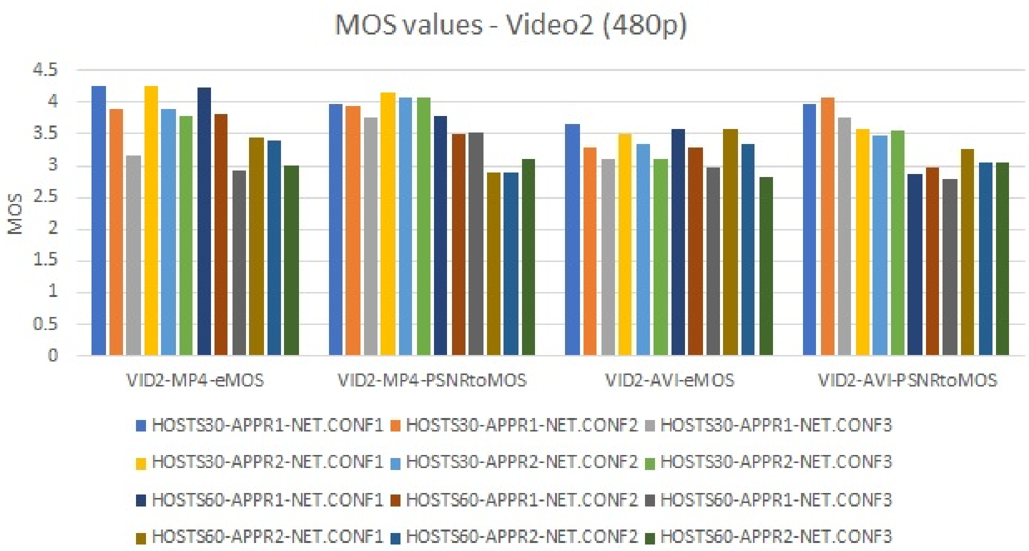


Figure 7.5.13: Average MOS estimation for video 2 when the resolution is 480p.

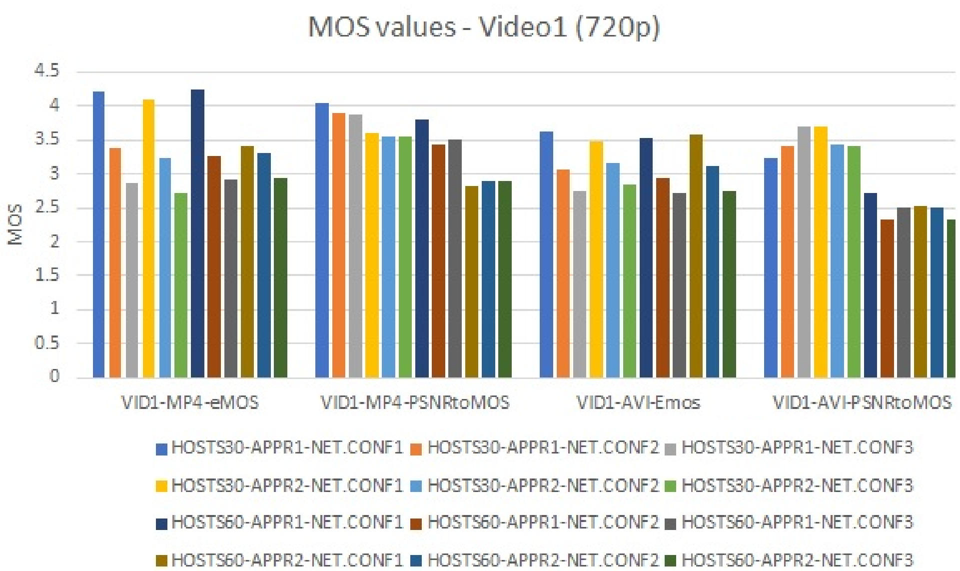


Figure 7.5.14: Average MOS estimation for video 1 when the resolution is 720p.

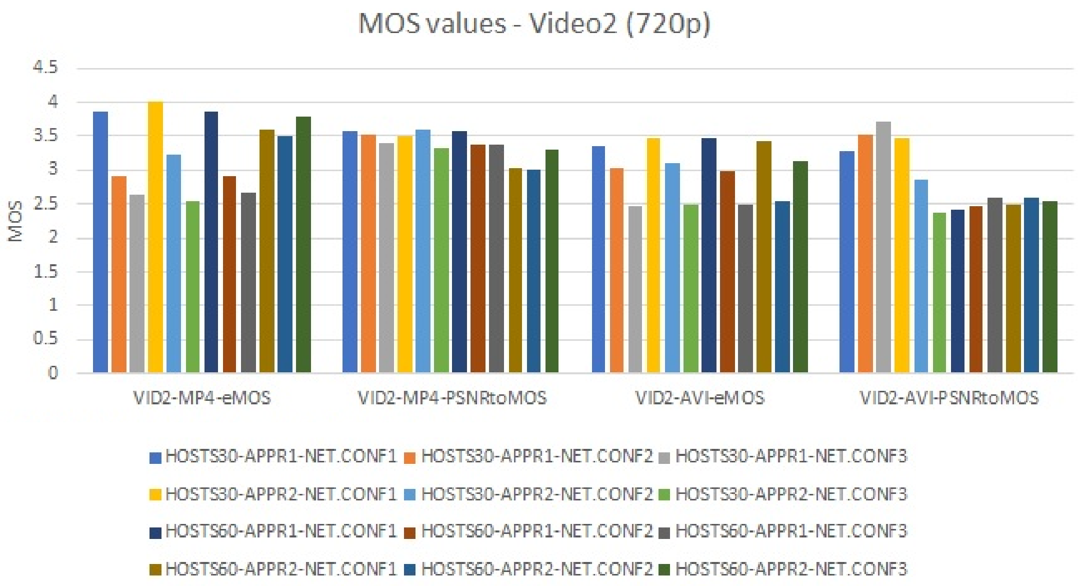


Figure 7.5.15: Average MOS estimation for video 2 types when the resolution is 720p.

Chapter 8

An SDN-approach for QoE management of multimedia services using resource allocation

8.1 Introduction

New heterogeneous requirements will accompany future networks in terms of end-users Quality of Experience (QoE) due to the increasing number of application scenarios being deployed. Network softwarization technologies such as Software Defined Networks (SDNs) and Network Function Virtualization (NFV) promise to provide these capabilities. In this work, a novel QoE-driven resource allocation mechanism is proposed to dynamically assign tasks to virtual network nodes in order to achieve an optimized end-to-end quality. The aim 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 service in question is divided into tasks, and the neighbour nodes negotiate the assignment of these considering the final quality. In work, I specifically focus on the video streaming service. I also show that the agility provided by SDN/NFV is a critical factor for enhancing video quality, resource allocation and QoE management in future networks. Preliminary results based on the Mininet network emulator and the OpenDaylight controller have shown that my approach can significantly improve the quality of a transmitted video by selecting the best path with normalized QoS values.

8.2 Related Works

The network resource allocation is a significant problem in multimedia communications. There have been many studies in recent years, tackling this problem. For example, [212] addressed the issue regarding the optimization of network resource allocation for wireless video delivery. With a focus on throughput maximization, a cross-layer QoE-based optimization framework was proposed in [213] to allocate network resources efficiently for a video delivery service.

In real-time applications such as video streaming, there are dynamic changes in the application requirements (e.g. delay, packet loss). Note that, initially, the QoE-based optimization solutions were proposed for elastic application such as file transfers, which captures the user‘s satisfaction as a function of data rate using a concave utility function [214]. A network operator‘s view aims to allow the maximum number of users to join the system and at the same time to keep a good level of service quality. Their technique is based on the fact that the user is choosing a charge per unit time, and the network is determining allocated rates. However, a balanced must exist between them in order to have an optimized system.

In [215], a framework is proposed which applies dynamic traffic shaping on home network gateway based on network traffic statistics and monitoring of video flow to achieve a dynamic allocation of bandwidth for each video flow in real-time. The authors in [216] present an SDN-testbed along with an SDN-based video streaming architecture for monitoring the streaming flows in real-time using MPLS protocol to change the routing paths. However, that approach is designed only from a video service provider perspective. In [217], the architecture that integrates the SDN and NFV is described along with the mandatory requirements of adopting new technologies in mobile networks. Authors demonstrate the feasibility of SDN and NFV technologies for future networks such as 5G. However, as the authors point out, robustness and reliability are not provided. In order to proceed with my work, I considered the work from [218]. By leveraging the NFV and SDN in order to solve the challenges of resource, traffic and mobility management in the current mobile networks such as 4G/LTE networks, new concepts and opportunities for the Software-Defined Mobile Network (SDMN) architecture is analyzed theoretically and experimentally in [218] to meet the end user's traffic requirements. In [219], authors demonstrate an SDN/NFV enabled network domain approach towards providing an agile video transcoding process for maintaining the QoE level of a media service when network congestion occurs. This approach adopts the encoding characteristics provisioned video stream to implements a self-optimization and self-adaptation VNF of real-time video streams. The authors in [220] have proposed a Video Control Plane in order to monitor the QoE delivered by a CDN that belongs to its pool and select the most appropriate when a new video request is received. Their solution predicts continuously the CDNs' performance based on clients' feedback and computed using the k-NN algorithm. A resource allocation architecture with automated QoE assessment is proposed in [221], which is based on affective computing and sensing and takes into account also a mixed context (licensed and license-exempt technologies).

8.3 Problem Formulation

At the network level, in the traditional systems, all media flows from the server to clients 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 [222]. Thus, it is required to develop mechanisms that allow delivering each user media flow over the "best available" path using the "best service configuration" to maximize the end-users' QoE. To achieve the QoE maximization, in my approach, I aim to provide QoE-centric traffic flow control and routing mechanisms, a concept which is motivated by the Economic Traffic Management presented in [223]. The QoE-centric traffic flow control and routing mechanisms aim to enable several network elements to cooperate in measuring and collection of the QoE influencing factors in the SDN network.

In this work, my target is to optimize the overall QoE in case of adaptive video streaming application using NFV and SDN technologies. The quality optimization problem consists of maximizing a quality function. I formulate the problem of dynamic task allocation in an adaptive video streaming scenario considering QoE influence factors in order to improve the overall QoE. To achieve this, I proposed a task model, a network model and a task assignment model, as described in section 4.2.

8.4 Task Assignment Algorithm

Using the network topology, shown in Figure 8.5.1 and the task assignment model from 4.2.2, my 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 the amount of CPU and memory. The resource allocation algorithm employs the following steps:

1. Based on network topology, the 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.

2. Based on the proposed network model and task model and the previous list, it creates a new list with all possible paths considering all the constraints. Moreover, every path must starts from a "media serve" node and ends with a "client" node and must include nodes that execute all the tasks.

3. For every path, it calculates the $QoS(C)$ value of the path based on Eq. 4.17. Since every link of the path in my topology has different delay and bandwidth, the algorithm considers the average delay and average bandwidth of the path.

4. For every path, then it calculates the "Benefit of the path" based on Eq. 4.16 by considering the $QoS(C)$ value of a path, Q_r (a factor which determines the

overall QoE of the video streaming), the resolution class A and the structure of the video frames R.

5. For every path, it calculates the "Cost of a path" by considering the required amount of CPU and memory of a task λ.

6. The algorithm calculates the u_{net} function, based on Eq.4.14 using the "Benefit of the path" and the "Cost of a path".

7. The algorithm will use the path with the highest u_{net} value to deliver the video to the Client.

8.5 Performance Evaluation

8.5.1 Experimental Setup

In my experimental testbed, I based on a network emulator called Mininet [182] and an SDN-Controller implemented by OpenDaylight [183]. The testbed consists of 2 video streaming servers, three clients, seven virtual switches and the SDN controller, as illustrated in Figure 8.5.1. 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.

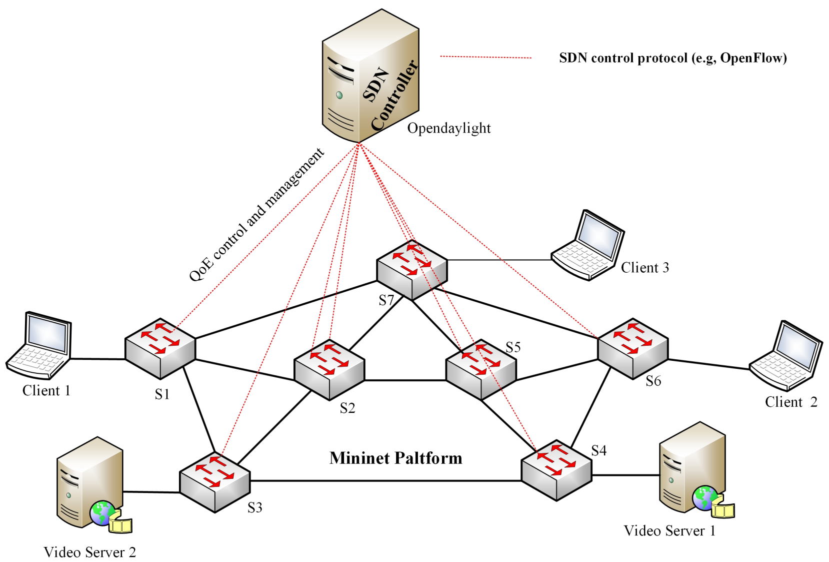


Figure 8.5.1: Experimental Setup

For simplicity, during network path assignment to a particular multimedia traffic/session flow, my approach first specifies the particular network path links and nodes by utilizing an automatic optimal path configuration of each traffic/session flows to be established through the assigned path loss probability and an average end-to-end delay of the link. During multimedia service flow establishment between a client and a video server, each of the following components performs the following functions [224]:

1. End-user clients initiate an SDN application request to the video server with their preferences and requirements such as video and screen resolution as well as the supported codecs of their devices.

2. Upon receiving a request, the video server (s) communicates these clients' requests to an SDN application with the parameter matching function. It also conveyed the client's information and requested parameters such as required video resolution, video bitrates and codecs.

3. An SDN application then determines the required parameters to be delivered to clients when there is a match received from the video server.

4. The associated multimedia 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. The SDN-Controller through the OpenFlow [27] 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.

When scalability and interoperability become the core requirements, I can create a generic solution using OpenFlow which should work across different service provisioning scenarios ranging from a multitude of vendors and ISPs.

8.5.2 Results

The objective of my experiments is to evaluate the QoE level provided by the proposed task allocation algorithm with respect to scalability and clients differentiation, bandwidth fluctuations and end-to-end delay variations. The programmability and the overall combination of features provided by SDN and NFV, as well as the openness of OpenFlow [27], enabled us to have a full realization of the real-world networks throughout my experiments. My experiments employ the VLC media player [225] for video streaming while for video stream delivery from any of the video server nodes to clients; the UDP/RTP protocols are used. The total available bandwidth to be accessible to video streams is set on different links to fluctuate between 20kb/s to 20Mb/s. Such 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. As shown in Table 8.5.1, the resolutions of video streams to be delivered to clients were selected randomly between 360p, 720p and 1080p at 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. In order to evaluate my 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 distribution was selected. The uncompressed YUV video files in 360p, 720p and 1080p resolution were then encoded using the H.264 codec. Three different tests were carried out to evaluate the performance of my approach. The first test evaluates the bandwidth fluctuations and delays variations from a video server to a certain number of receiving clients. In the second test, I conducted two different experiments 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 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 Eq. 4.17.

Table 8.5.1: 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

The network QoS parameters (packet loss rate, jitter, delay and bandwidth) which are also related with video quality were configured using netem [226] which is the well-known routing and traffic control feature for system monitoring, traffic classification and traffic manipulation.

Bandwidth and end-to-end delay variations

Figure 8.5.2 shows the relationship between the number of video servers that can serve a specified number of clients at varying link bandwidth and delay variation values. In practice, a change in the shared bandwidth will lead to a network resource reallocation process which is instructed by the 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.

I observe that, as the number of video delivery nodes increase, 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 h[h]

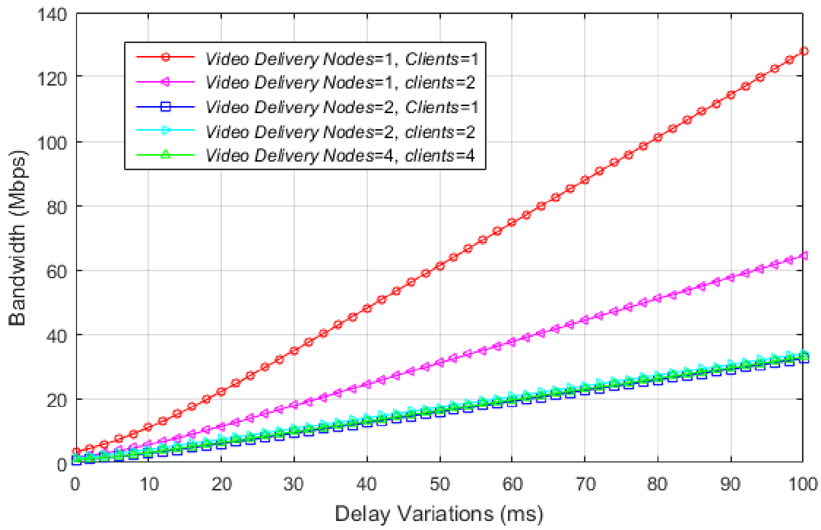


Figure 8.5.2: Bandwidth and delay variations with different number of clients and video servers.

services to 1, 2 or 4 clients at the same bandwidth of 20Mbps. Such observation also implies delivering high videos quality to clients.

Effects of Packet Loss variations on video quality

As multimedia applications over IP networks continue to gain popularity, resource allocations with respect to multi-tier topology, user data sharing and cross-domain policies to be implemented using SDN approach continue to face a challenge as well [227]. Considering the media streaming services in dynamic and heterogeneous applications in future networks such as 5G, my approach differs from the conventional designs in the sense that, I model the utility and assignment of tasks 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, I conducted two different experiments 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 reason enough for my experimentation taking into account that the aim was to investigate how delay and packet loss affect the transmitted video quality using my approach. Figure 8.5.3 shows the video quality of transmitted videos as the function of the packet loss rate.

As expected, I observe from Figure 8.5.3 that, as packet loss decreases, the video quality increases as indicated by the QoE values from the correlation model described in [97].

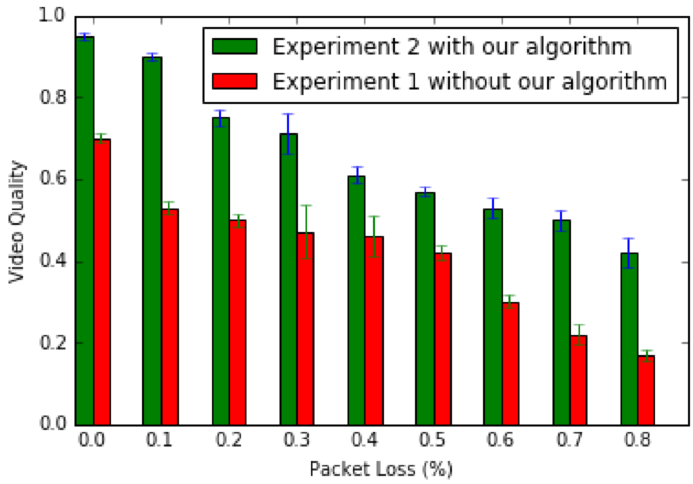


Figure 8.5.3: QoE values at different variations of packet loss and delay

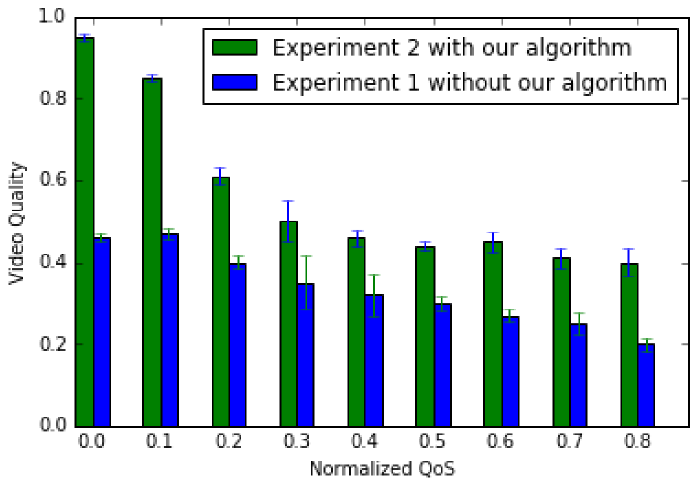


Figure 8.5.4: Transmitted video QoE with the normalized QoS values.

Transmitted video QoE with the normalized QoS values

Figure 8.5.4 shows the results of the transmitted video QoE without and with my proposed QoE-driven resource allocation algorithm. In this experiment, the normalized $QoS(C)$ was calculated based on Eq. 4.17 where the constants A and R in Eq. 4.16 were assigned to 240 and 24, respectively by considering the used codec, network parameters set and video resolution. The blue colour bar demonstrates the test experiment without using my proposed algorithm. The video quality increases as the normalized QoS value decreases. The video quality and the $QoS(C)$ reflect the QoE metric for video streaming services and the network parameters/conditions, respectively. They were described in the second test done in the previous subsection. Using the QoS-to-QoE correlation model in [97], my proposed approach can significantly improve the video quality at the normalized QoS values.

8.6 Conclusion

Although SDN and NFV promise new opportunities, a unified approach for leveraging these network softwarization technologies in the wireless and mobile domain is lacking. This work presents an SDN-based approach for QoE management through cooperation and information exchange among network elements which are involved in the service delivery chain (e.g., from the video delivery nodes to clients). This approach enables more efficient resource utilization and simplifies network management using the elasticity of SDN/NFV technologies. I aim to find and provide the best combination of network nodes that can cooperate during the execution of the defined task and at the same time to improve the overall QoE level of the end-users. To achieve this, I developed a QoE-driven dynamic task allocation scheme for adaptive video streaming over SDN/NFV enabled networks. I have shown that the agility provided by network softwarization infrastructures using SDN/NFV is a key factor for enhancing video quality, resource allocation and QoE management, especially in future networks. In the future, I will replicate this study to achieve an efficient QoE control and management of network/system resources between multiple players in the network domain (mobile network operators, content providers and cloud/service providers.

Chapter 9

A QoE monitoring solution for multimedia services in 5G networks

Introduction

Network Function Virtualization (NFV) and Software Defined Network (SDN) paradigms shift towards virtualization [53]. Without the virtualization technology, it is challenging to monitor a virtual network with a traditional (physical) monitoring probe since the traffic in the virtual networks is invisible to the physical network [62]. In a virtualized environment, the traffic is transmitted via virtual functions among Virtual Machines (VMs). Since these data are not leaving the VM, the traditional (physical) monitoring probes cannot monitor them. This leads to a significant challenge regarding the **type** (traditional or virtual) of the monitoring probes. In contrast, the virtual monitoring probes, in the form of Virtual Network Functions (VNFs), can communicate with physical and virtual network devices in order to monitor the QoE level in both environments.

Furthermore, the **placement** of the monitoring probes (virtual or physical) by the ISPs remains as a challenge [104] [228]. Because the monitoring technique that has been chosen should conduct accurate measurements [228] and at the same time decrease the energy consumption, the computational complexity and the deployment cost [229].

Another challenging task regarding the QoE monitoring in 5G networks comes from the fact that there is a **diversity** of devices and a variety of data considering the plethora of new emerging services such as HDTV,3DTV, video streaming and Social TV. Accordingly, the different services have different requirements in terms of QoE [230] and thus, a flexible QoE monitoring solution is needed in order to support the diversity of services.

In the current work, I propose a monitoring solution for 5G networks which includes QoE agents that have been introduced in [8] in the form of Virtual Network Functions (VNFs) in order to monitor user's satisfaction without increasing network load and the complexity during the QoE monitoring. The QoE agents are deployed in different network locations, and they can collect QoE related parameters and based on them to estimate the QoE level using the most appropriate QoE models for each service type. In this work, I make a step further by relocating the QoE agents from their conventional places in order to achieve economic benefits and improve the flexibility and the scalability of the monitoring solution.

I propose a QoE agent placement solution in order to monitor the QoE level for a variety of services. In this paper, I propose a model to optimize the QoE agent's performance. This can be done by the proposed algorithm, which efficiently selects the lowest number of monitoring locations that do not increase the computational complexity and the network load during a QoE monitoring. In this work, I focus on an adaptive video streaming service. The QoE Agents support the diversity of services, and they can be used for any service using a related QoE estimation model. I formulate the problem of QoE agent placement problem in a "QoE-monitoring" scenario for an adaptive video streaming service. Preliminary results show that the most cost-effective and efficient approach is to follow a distributed approach and deploy the QoE agents in NFV-capable network devices and the Cloud. Moreover, by following this approach, I can leverage from the real-time insights. Furthermore, when the QoE agent is deployed in the Cloud, I observe that I achieved my target because the network load is decreased as much the computational complexity in terms of CPU utilization and memory utilization.

9.1 Related Works

Most of the past works focused on improving the Virtual Machine (VM) performance, and availability, scalability, energy consumption and VM live migration cost [231]. Thus, the placement decisions are based on the following, (a) the resource availability (computational power, storage, bandwidth), (b) the deployment cost and (c) the flexibility of management and maintenance. Due to a potential high load of the virtual management systems, a trade-off has to be made between accuracy and cost-effective deployment. It is critical to base any decisions on accurate QoE monitoring and management.

In the past, a study [228] has been conducted, but it was application-specific (only online video service in the network). In [228], the authors design a VNF monitoring system to measure the video quality and estimate the QoE at the client-side. They place the VNF in two locations, nearby and far away from the user in order to analyze the impact of geographical placement of the VNF on its performance. They showed that the VNF had high accuracy in QoE estimation if it was deployed at the edge network close to the user and lower accuracy when it was deployed in a Data Center. However, many advantages can exist when a QoE monitoring VNF is deployed in a Data Center, such as the storage and analysis capabilities and real-time insights.

In order to succeed agile network management, the SESAME [232] project develops and demonstrates an architecture to promote the Cloud-Enabled Small Cells (CESCs) which supports edge cloud computing in a multi-tenant and multi-service ecosystem with a target to support enhanced edge cloud services by enriching small cells with microservers. By the deployment of edge cloud services, it is possible to benefit from the fact that they are deployed near to the user. Thus more accurate measurements can be done with fewer delays. The ANYaaS [12] concept is introduced for the 5G mobile networks and relies on NFV and Cloud computing in order to manage the mobile services efficiently and optimize the usage of network resources. Using any IaaS, the mobile operators can manage and orchestrate multiple service instances dynamically using Docker virtualization approach. Efficient network resources allocation should be applied in order to decrease energy consumption and costs.

In the context of cloud computing, a set of optimization metrics can be defined [233] such as energy consumption, latency minimization, QoE maximization and number of migrations optimization, which are suitable for MEC systems even though they have lower performance and power consumption due to their limited capabilities (in comparison with Cloud Data Centers).

In [234] the QoE monitoring as VNF is deployed in provider edge without to consider the service type or the factors that affect this placement [234]. The ISPs should place the monitoring probes in the form of VNFs by considering the cost and the effectiveness of [104]. The decision where to place a service can be based on a variety of criteria, such as hardware requirements, network latency between the service instance and the client, deployment and operating costs of the service, The network provider should find the balance between these objectives when deciding where to deploy their services.

The minimization of resource utilization is an important problem that has been studied by [235], [236], [237] [15]. The authors in [235] dealt with the VM placement problem and proposed an efficient online algorithm that can be applied in a dynamic environment under a variety of traffic loads in order to minimize resource utilization. An online heuristic algorithm [236] achieves the optimal resource allocation in a Cloud-based environment in order to minimize resource utilization and reduce the VM migration overhead. A framework [237] minimizes the end-to-end latency by using a fast algorithm to implement a random cloud selection (to optimize resource utilization) and heuristics using the Support Vector Regression machine learning technique to find the optimal placement. A placement and provisioning optimization strategy [15] in an Edge-Central Cloud that takes into account the QoS requirements optimizes the resource utilization, prevents Cloudlet overload and avoids violation of QoS requirements. In congestion with the work of [15], a trade-off found [238] between the resource utilization and violation of SLA requirements while improves the user experience and ensures scalability.

About the minimization of network resource consumption several approaches have been proposed [16], [17], [239],[240] [241][242].

In a Cloud-based environment, [16] [17] VNF placement approaches are proposed to minimize it while considering the network traffic and the service requirements. A trade-off between the minimum traffic and the minimum number of VNF instances is achieved [239] using a heuristic deployment algorithm. In addition to the network resource utilization, the minimization of computational resources is also achieved by three optimization models [241](single and multi-objective) that consider latency requirements and three heuristic algorithms [242] (online and offline scenarios) that consider resource and traffic requirements. Furthermore, user experience is considered by the CDN owners in order to provide better services [243] while minimizing the operating costs. A trade-off is achieved [37] when active probes are deployed in a widely distributed heterogeneous Cloud environment before the service deployment.

Several studies have been conducted on the service placement while considering the deployment cost [244][245][246][34][247]. An Intenet-of-Things (IoT) service is deployed on FOG environment [247] taking into account QoS requirements and a Cloud-based framework that applies a Cloudlet placement [246] that uses two models. The one model considers that the network status is static and the other or that it is dynamic in terms of load variation and user's mobility. Also, in a Cloudlet placement, the end-to-end delay is considered [34] in order to find the optimal locations for Cloudlets while reducing the number of them and their servers. Moreover, the optimal solution can be found in a shorter time by supporting service differentiation between the users [245] without compromising the QoS requirements. By placing the service in the network Edge [248] can be achieved the optimal allocation with the minimum end-to-end latency between the users and network functions. By using an heuristic-based allocation mechanism in a MEC environment [249**?**] to dynamically perform resource utilization and execute a re-optimization algorithm periodically to adjust allocation can avoid the increment of the operating costs and keep the latency in the minimum level. An optimization module [235] can be used to take high-quality placement decisions based on current network conditions while considering the energy consumption.

Thus, I propose a QoE monitoring solution for 5G networks by deploying QoE agents in strategic locations. The ISPs can reduce the monitoring overhead, and increase their network scalability and reliability. I propose the deployment of the Slave agents in the user's premises in order to keep latency in minimum level and the Master agent should be deployed in a MEC server or in a Data Center in the Cloud in order to leverage from their processing capabilities and the real-time insights.

9.2 The QoE-based placement of the Agents

In [8], the QoE agents applied in specific network entities for QoE estimation by considering the network load and the estimation accuracy. For each layer, different parameters are retrieved, different processes are executed, different inputs and outputs are defined from different network components. The QoE agent in [8] follows the "Distributed Agent" approach and deployed in an LTE-Advanced Pro network and implements a "Slave agent" and one "Master agent". The "Master-Agent" is deployed in the Packet Gateway (PGW) because it has a direct connection with the Policy Charging Rules Function (PCRF) entity and with Internet, e.g. Over-The-Top (OTT) providers and, the "Slave agent" is deployed in the PCRF entity because it has direct connection with other useful network entities. The "Master agent" communicates with the "Slave agent" and with the external applications, such as QoE-aware application, on the Internet side. This approach was described in Section 7.3.1.

In this section, I propose a QoE agent placement model which is based on a resource allocation approach proposed in the models from 4.2. My target is to reduce the complexity and the traffic related to the QoE monitoring procedures in a 5G network by using distributed agents. The optimization problem consists of maximizing a quality function. I formulate the problem of dynamic QoE agent placement in a "QoE-monitoring" scenario by conducting a lighter QoE monitoring. To achieve this, I used the task model from 4.2.2, and I proposed a network model that is described in 4.5.

9.2.1 Placement approaches

I propose that each task (MQA and SQA) should be executed in the NEs by following one of the below approaches, as shown in Figure 9.2.1.

1. The MQA is deployed in the Core Network, and the SQA deployed in virtual Customer Premises Equipment (vCPEs). This approach is based on [8], and it follows the proposed QoE agent placement. The vCPE is a software-defined CPE that delivers network services, such as routing, and VPN functions and this approach is especially beneficial for businesses where investments of dedicated hardware equipment and their costs like power consumption and hardware maintenance, could be considered burdening. The ISPs leverage the benefits of vCPEs due to the fact that they can deploy simple and fast new services, reduce the management complexity, and reduce OPEX/CAPEX [250]. Also, the ISPs can benefit from the deployment of the QoE agent in a vCPE because they will have reliable and consistent visibility into the customer experience.

2. The SQA is executed in the vCPEs and the MQA in a MEC entity. By leveraging

the benefits of Edge Computing, I deploy the MQA in a MEC server because it has more processing capabilities. The MQA needs more resources in order to be able to conduct the QoE estimations.

3. The SQA is executed in the vCPEs and the MQA in the Cloud. Deploying the MQA in the Cloud, I can take advantage of its storage and real-time analytics capabilities. The benefit of the MQA deployment in the Cloud is that I can get real-time insights.

In the above approaches in most of the cases the SQA is executed in a lower level in the network (e.g. in Customer's side) due to the fact that it needs less computational resources in comparison with the MQA which needs more because it includes the QoE estimation model.

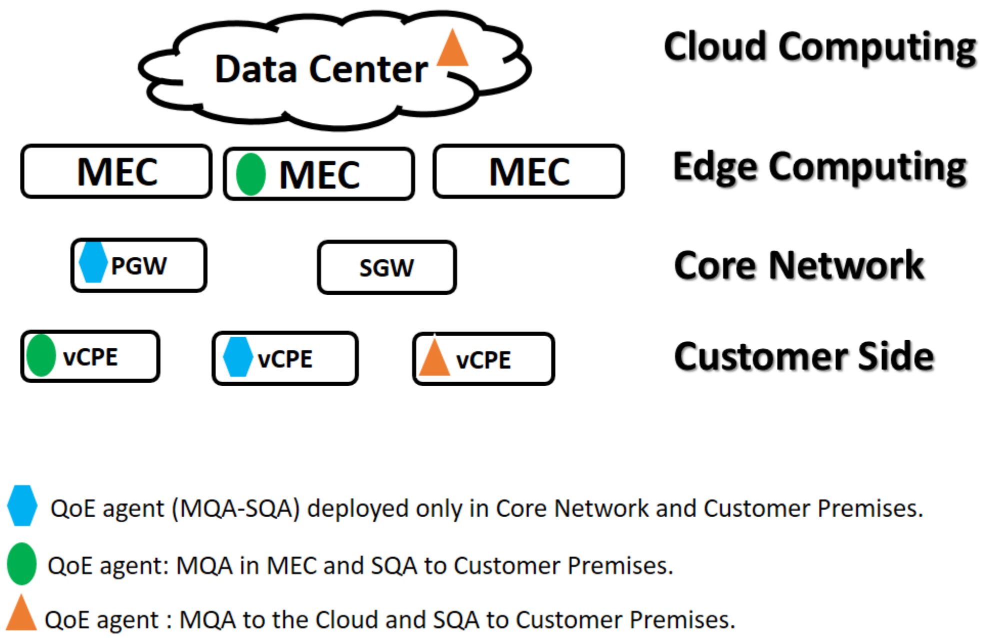


Figure 9.2.1: Network model and placement approaches

The aim is to propose a lighter approach for QoE monitoring, considering the deployment location of the QoE agents. This problem could be time and energy-consuming if I have many services and massive traffic. I apply a distributed optimization approach to make the most efficient QoE agent placement in order to decrease the overhead of QoE monitoring.

9.2.2 QoE agent placement algorithm

Using the network topology shown in Figure 9.2.1 and the estimation model from section 4.5, my algorithm finds which is the best location to install the monitoring agents (MQA, SQA) in order to decrease the complexity and improve the performance of the QoE monitoring. The proposed algorithm efficiently selects the lowest number of monitoring locations (lower costs). Every NE has specific network resources, and every task (QoE agent) requires a specific amount of resources regarding the amount of CPU and memory. The location-aware placement algorithm employs the following steps:

1. Based on network topology, the algorithm finds the VNF-able nodes that can be used to deploy a QA (MQA or SQA) and creates a list of them.

2. In that list identify if a VNF-able node can execute an MQA, an SQA or both. If it has the available computational resources.

3. Based on the proposed network and task model and the previous list, it creates a new list with the possible locations considering the constraints. The combinations of possible locations are described in Figure 9.2.1.

4. For every path with locations which are related with the proposed placement approaches in Figure 9.2.1, it calculates the $c_{i_\delta^\lambda}$ value by considering the required amount of CPU and memory of a task λinT. The path includes locations/NEs that a QA can be deployed.

5. For every path with a set of locations, the algorithm calculates the $b_{i_\delta^\lambda}$.

6. The algorithm calculates the aac_{net} function using the $b_{i_\delta^\lambda}$ and the $c_{i_\delta^\lambda}$.

7. The algorithm will use the path with the highest acc_{net} value to monitor the QoE in the network in order to reduce the computational complexity of QoE monitoring.

9.3 Performance evaluation

9.3.1 Experimental Setup

The proposed solution was implemented on the Mininet emulation environment [251]. An HTTP apache server was used to store the video content. The video streamed was the "Big Buck Bunny" encoded in 806kpbs. The HAS clients were implemented on top of the libdash library [252]. Opendaylight [253] controller was used and Open vSwitches [254] was used to realize the OpenFlow switches. The emulated network is shown in Figure 9.3.1, where the locations of the QoE agents are illustrated. In order to provide an extensive evaluation of the proposed solution, I emulate 30 episodes of the video trace and average the results over the 30 runs for each placement approach. The average bandwidth in the emulated links was 3 - 10 Mbps, the packet loss was 1 - 5% and the jitter 5 - 20ms [255] [98].

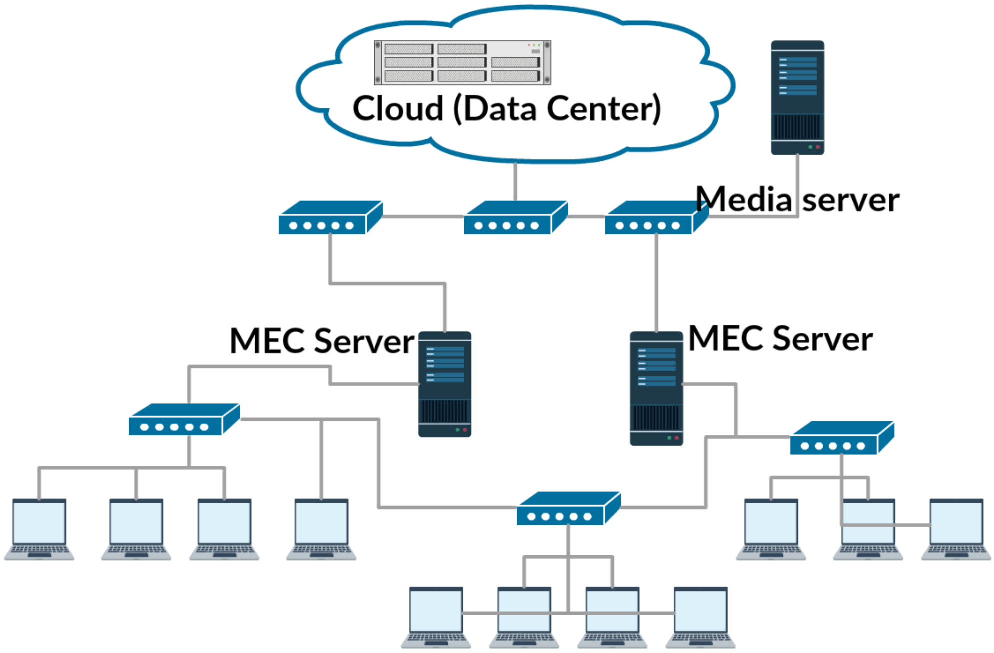


Figure 9.3.1: Emulated topology

The objective of the performance evaluation was to present for an adaptive video streaming service which QoE agent placement approach can reduce the complexity and overhead of the QoE monitoring. I used two different QoE estimation models for an adaptive video streaming service.

I evaluate my model and the algorithm in 3 cases (3 placement approaches), as shown in Figure 9.2.1, during an adaptive video streaming service:

1. **Approach 1**: When the MQA is deployed in the Core Network (as in [8]), and the SQA is deployed in Customer's side (in a vCPE).

2. **Approach 2**: When the MQA is deployed as edge cloud service in a MEC server, and the SQAs are deployed in the Customer's side (in a vCPE).

3. **Approach 3**: When the MQA is deployed in a Data Center in the Cloud, and the SQAs are deployed in the Customer's side (in a vCPE).

9.3.2 Results

Network load vs QoE agent placement algorithm

For each placement approach, I observed the network load with and without the QoE agent placement algorithm. In Figure 9.3.2, I can observe that the network load in case the QoE Agents are deployed in the Customer's Premises (Slave agents) and the Master Agent in the Cloud is lower due to the fact that the Slave agents are placed in strategic

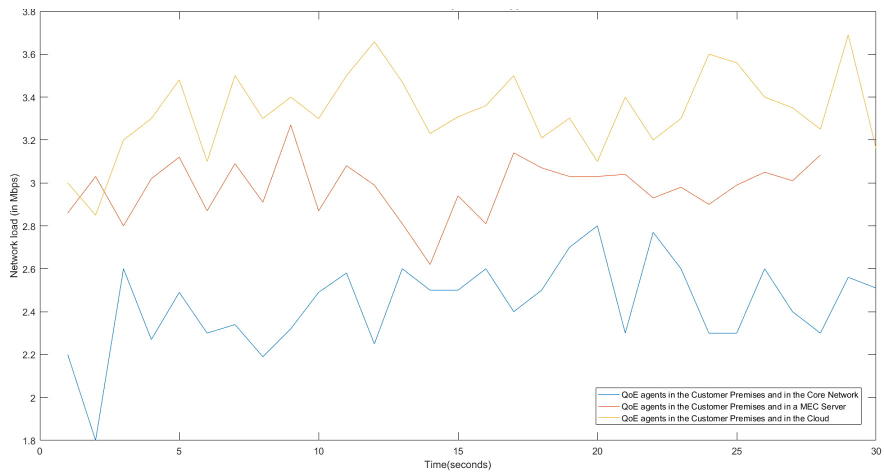


Figure 9.3.2: Network load for each QoE agent placement approach

locations in the ISPs' network. Thus, they can forward their information directly to the Master agent without increasing the network load.

Resource utilization vs QoE agent placement approaches

For each placement approach, I observed resource utilization in the form of CPU and memory utilization. In Figure 9.3.3, I can observe the CPU and the Memory utilization

Figure 9.3.3: Emulated topology

for every placement approach. It is observed that when the Master QoE agent is deployed in a Data Center in the Cloud is not consuming many computational resources as happened in case the Master QoE agent is deployed in the Core Network. The VNF-able devices in the network do not have as many capabilities as a Data Center. Also, by the implementation of the Master Agent in the Cloud, I can have real-time analytics and insights about the QoE level using analytics engines such as Elastic Search [187].

9.4 Conclusion

In this work, I present a light QoE monitoring solution for 5G networks. My target was to identify the optimal locations to deploy the QoE agents in order to decrease the network load and the computational complexity during the QoE monitoring procedure. I proposed three placement approaches, and I evaluated them in order to find the most efficient placement of the QoE agents. A task model and a network model presented in order to formulate my placement problem. During my experiments, I observed that each QoE agent deployment had its advantages and disadvantages in cost, latency, resource usage. The preliminary results show that the more flexible and cost-effective approach was to deploy the QoE agents in NFV-able network devices in the user's side and Data Centers in the Cloud.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

In this thesis, the concepts of Quality of Experience regarding QoE monitoring and management are studied where new approaches compared to the state-of-the-art were proposed. In Chapter 2, I described the State-of-the-Art and the main concepts used in this thesis.

In Chapter 3, I described the State-of-the-Art regarding the concept of Virtual Probes (vProbes).

In Chapter 4, I described the QoE estimations models that were used during this thesis.

In Chapter 5, I described the concept of vProbes. I identified and described the vProbe-based approaches, and I have observed that vProbes are preferred in comparison with physical probes because they are more flexible, adaptable and cheaper. Nevertheless, the hybrid approaches are the optimal solution to deploy a vProbe, because most of the network providers are under a "virtualization technology" adoption phase and in hybrid approaches the physical probes can be used in cooperation with vProbes. Considering the vProbe's characteristics and the current technologies used by network operators, I observed that exist several challenges regarding their deployment, such as the performance of the deployment environment and the collaboration with the physical infrastructures.

In Chapter 6, I described a QoE monitoring and management solution that includes a framework and an SDN-based architecture. The results from the performance evaluation had shown that eMOS and its parameters are related to the network conditions, the video resolution and the quantization parameter.

In Chapter 7, I described two Agent-based solutions for QoE monitoring. I employ distributed monitoring agents, called "QoE Agents", in the form of Virtual Network Functions (VNFs) that monitor different QoE influence parameters. From the performance evaluation, I realized that the dynamic adjustment of monitoring frequency has a significant impact on the accuracy of the QoE estimations.

In Chapter 8, I described an SDN-approach for QoE management of multimedia services using resource allocation. I proposed an SDN-based approach for QoE management through cooperation and information exchange among network elements which are involved in the service delivery chain. I have shown that the agility provided by network softwarization technologies is a key factor for enhancing video quality, resource allocation and QoE management.

In Chapter 9, I described a QoE monitoring solution for multimedia services in 5G networks. From the performance evaluation, I observed that the more flexible and cost-effective approach was to deploy the QoE agents in NFV-able network devices in the user's premises and Data Centers in the Cloud.

10.2 Future Work

In future work, I aim to propose a solution in order to monitor the QoE level of a "Social TV" application using sentiment analysis. The basic idea is to create a model based on Sentiment Analysis in order to retrieve the subjective impression for Social TV Applications. The main goal is to monitor the feelings of the User. Based on my knowledge, there is no model in the application layer, which uses Sentiment Analysis to retrieve subjective impression.

Smart cities have been identified as a strategy to decrease the problems caused by the explosive population growth in metropolitan areas, and as a consequence, it has the explosive increment of energy consumption [256]. In this context, smart homes can provide decision support tools to assist the users to make cost-effective decisions when utilizing electrical energy [257]. Moreover, Smart home has been identified as one of the critical applications to improve people's lifestyles, and it is bringing a great deal of attention in academia and industry [258]. Technologies that used in a Smart Home environment for energy saving and management are the smart plugs, the occupancy sensors [259]. In the context of a Smart Home by monitoring and controlling the energy consumption, the user's benefit from the fact that they can decrease their energy consumption and consequently, their energy cost. Many people have not experienced the comfort that Smart Homes should have brought into their lives [258]. The customer comfort can be defined as a set of constraints on appliance usage, a priori set without profiling among different kind of customers that have different needs[257]. The comfort perceived by customers when policies are applied outstanding because consumers' acceptance is needed to improve their Quality of Experience (QoE). My target is to monitor QoE in terms of comfort in a "Smart house" environment in order to improve user's satisfaction

Furthermore, I am going to present a QoE-based overview on Tactile Internet in order to identify how I can improve the QoE and avoid cybersickness. Considering the main QoE challenges which are (i) to keep latency and jitter in required limits demands for new mechanisms when taking the movement decision and (ii) monitor and manage of real-time services such as tactile internet, ar/VR, gaming and mission-critical services.

Appendix A

Bibliography

[1] Ericsson, "On the pulse of the networked society,," 2016.

[2] G. Gómez, Q. Pérez, J. Lorca, and R. García, "Quality of service drivers in lte and lte-a networks," *Wireless personal communications*, vol. 75, no. 2, pp. 1079–1097, 2014.

[3] P. Le Callet, S. Möller, A. Perkis, and et. Al., "Qualinet white paper on definitions of quality of experience," *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, vol. 3, 2012.

[4] Y. Wang, P. Li, and et. Al., "A data-driven architecture for personalized qoe management in 5g wireless networks," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 102–110, 2017.

[5] J. Matias, J. Garay, and N. e. Toledo, "Toward an sdn-enabled nfv architecture," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 187–193, 2015.

[6] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc)," *IEEE Network*, vol. 28, no. 6, pp. 18–26, 2014.

[7] A. Takahashi and T. Hayashi, "Collaborative quality framework: Qoe-centric service operation in collaboration with users, service providers, and network operators," *IEICE Transactions on Communications*, 2017.

[8] E. Grigoriou, T. Saoulidis, L. Atzori, V. Pilloni, and C. Periklis, "An agent-based qoe monitoring strategy for lte networks," in *Communications (ICC), 2018 IEEE International Conference on*, pp. 1–6, IEEE, 2018.

[9] E. Grigoriou, T. Saoulidis, L. Atzori, V. Pilloni, and P. Chatzimisios, "A qoe monitoring solution for lte-advanced pro networks," in *2018 IEEE 23rd International*

Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–5, IEEE, 2018.

[10] A. Haidine and S. El Hassani, “Lte-a pro (4.5 g) as pre-phase for 5g deployment: closing the gap between technical requirements and network performance,” in *Advanced Communication Systems and Information Security (ACOSIS), International*, pp. 1–7, IEEE, 2016.

[11] H. Tullberg, P. Popovski, Z. Li, M. A. Uusitalo, A. Hoglund, O. Bulakci, M. Fallgren, and J. F. Monserrat, “The metis 5g system concept: Meeting the 5g requirements,” *IEEE Communications magazine*, vol. 54, no. 12, pp. 132–139, 2016.

[12] T. Taleb, A. Ksentini, and R. Jantti, “" anything as a service" for 5g mobile systems,” *IEEE Network*, vol. 30, no. 6, pp. 84–91, 2016.

[13] R. Cziva and D. P. Pezaros, “Container network functions: bringing nfv to the network edge,” *IEEE Communications Magazine*, vol. 55, no. 6, pp. 24–31, 2017.

[14] P. L. Ventre, A. Caponi, G. Siracusano, D. Palmisano, S. Salsano, M. Bonola, and G. Bianchi, “D-streamon: from middlebox to distributed nfv framework for network monitoring,” in *Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on*, pp. 1–2, IEEE, 2017.

[15] F. Ben Jemaa, G. Pujolle, and M. Pariente, “Analytical models for qos-driven vnf placement and provisioning in wireless carrier cloud,” in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 148–155, ACM, 2016.

[16] A. Gupta, M. F. Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee, “Joint virtual network function placement and routing of traffic in operator networks,” *UC Davis, Davis, CA, USA, Tech. Rep*, 2015.

[17] A. Gupta, M. F. Habib, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, “On service-chaining strategies using virtual network functions in operator networks,” *Computer Networks*, vol. 133, pp. 1–16, 2018.

[18] Y. Chen and L. Ruckenbusch, “Mobile edge computing: brings the value back to networks,” *IEEE Software Defined Networks Newsletter*, 2016.

[19] N. Bouten, R. d. O. Schmidt, J. Famaey, S. Latré, A. Pras, and F. De Turck, “Qoe-driven in-network optimization for adaptive video streaming based on packet sampling measurements,” *Computer networks*, vol. 81, pp. 96–115, 2015.

[20] J. Seppänen, M. Varela, and A. Sgora, "An autonomous qoe-driven network management framework," *Journal of Visual Communication and Image Representation*, vol. 25, no. 3, pp. 565–577, 2014.

[21] E. Liotou, H. Elshaer, and R. e. Schatz, "Shaping qoe in the 5g ecosystem," in *Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 2015.

[22] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[23] T. Wood, K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: integrating software defined networking and network function virtualization," *IEEE Network*, vol. 29, no. 3, pp. 36–41, 2015.

[24] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[25] S. Jain, A. Kumar, and et. Al., "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[26] A. Shawish and M. Salama, "Cloud computing: paradigms and technologies," in *Inter-cooperative collective intelligence: Techniques and applications*, pp. 39–67, Springer, 2014.

[27] N. McKeown, "Software-defined networking," *INFOCOM keynote talk*, vol. 17, no. 2, pp. 30–32, 2009.

[28] D. Kreutz, F. M. Ramos, and P. E. e. Verissimo, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[29] M. Jarschel, T. Zinner, and et. Al., "Interfaces, attributes, and use cases: A compass for sdn," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 210–217, 2014.

[30] N. F. V. ETSI, "Architectural framework," *ETSI GS NFV*, vol. 2, p. v1, 2013.

[31] ETSI, "Etsi ts 129 214 universal mobile telecommunications systems (umts; lte; policy and charging control over rx reference point," 2014.

[32] J. Costa-Requena, J. L. Santos, and V. F. e. Guasch, "Sdn and nfv integration in generalized mobile network architecture," in *European Conference on Networks and Communications (EuCNC)*, pp. 154–158, IEEE, 2015.

[33] T.-Y. Kan, Y. Chiang, and H.-Y. Wei, "Task offloading and resource allocation in mobile-edge computing system," in *2018 27th Wireless and Optical Communication Conference (WOCC)*, pp. 1–4, IEEE, 2018.

[34] Q. Fan and N. Ansari, "Cost aware cloudlet placement for big data processing at the edge," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.

[35] G. Orsini, D. Bade, and W. Lamersdorf, "Computing at the mobile edge: Designing elastic android applications for computation offloading," in *2015 8th IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 112–119, IEEE, 2015.

[36] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey," *arXiv preprint arXiv:1810.00305*, 2018.

[37] F. Vandeputte, L. Vermoesen, D. Griffin, T. K. Phan, M. Rio, P. Simoens, P. Smet, D. Bursztynowski, F. Schamel, and M. Franke, "Evaluator services for optimised service placement in distributed heterogeneous cloud infrastructures," in *2015 European Conference on Networks and Communications (EuCNC)*, pp. 439–444, IEEE, 2015.

[38] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[39] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[40] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[41] A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing: Principles, architectures, and applications," in *Internet of Things*, pp. 61–75, Elsevier, 2016.

[42] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Generation Computer Systems*, vol. 70, pp. 138–147, 2017.

[43] "Itu-t p.10. g.100 amandement 5: New definitions for inclusion in recommendation itu-t p.10 g.100."

[44] ITU-T, "10/g. 100 amendment 1,"new appendix i–definition of quality of experience (qoe)"," *International Telecommunication Union*, 2007.

[45] ETSI, "Etsi tr 102 643: Human (hf) quality of experience (qoe) requirements for real-time communication services."

[46] P. Le Callet, S. Möller, and A. e. Perkis, "Qualinet white paper on definitions of quality of experience," *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, vol. 3, 2012.

[47] "Itu-t g.1011: Reference guide to quality of experience assessment methodologies."

[48] "Itu-t p.800: Methods for subjective determination of transmaission quality.."

[49] S. Lee, K. Levanti, and H. S. Kim, "Network monitoring: Present and future," *Computer Networks*, vol. 65, pp. 84–98, 2014.

[50] Z. Su, T. Wang, Y. Xia, and M. Hamdi, "Flowcover: Low-cost flow monitoring scheme in software defined networks," in *Global Communications Conference (GLOBECOM)*, pp. 1956–1961, IEEE, 2014.

[51] D. Soldani, M. Li, and R. Cuny, *QoS and QoE management in UMTS cellular systems*. John Wiley & Sons, 2007.

[52] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger, "From packets to people: Quality of experience as a new measurement challenge," in *Data traffic monitoring and analysis*, pp. 219–263, Springer, 2013.

[53] 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, 2017.

[54] M. Liyanage, J. Okwuibe, I. Ahmed, M. Ylianttila, O. L. Pérez, M. U. Itzazelaia, and E. M. de Oca, "Software defined monitoring (sdm) for 5g mobile backhaul networks," in *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pp. 1–6, IEEE, 2017.

[55] L. Skorin-Kapov and M. Varela, "A multi-dimensional view of qoe: the arcu model," in *MIPRO, 2012 Proceedings of the 35th International Convention*, pp. 662–666, IEEE, 2012.

[56] "Speech and multimedia transmission quality (stq);quality of experience; a monitoring architecture."

[57] F. Espinet, D. Joumblatt, and D. Rossi, "Framework, models and controlled experiments for network troubleshooting," *Computer Networks*, vol. 107, pp. 36–54, 2016.

[58] G. Gheorghe, T. Avanesov, and M.-R. e. Palattella, "Sdn-radar: Network troubleshooting combining user experience and sdn capabilities," in *Conference on Network Softwarization (NetSoft)*, pp. 1–5, IEEE, 2015.

[59] A. W. Yusuf-Asaju, Z. B. Dahalin, and et. Al., "Mobile network quality of experience using big data analytics approach," in *International Conference on Information Technology (ICIT)*, pp. 658–664, IEEE, 2017.

[60] L. Atzori, A. Iera, and G. Morabito, "From" smart objects" to" social objects": The next evolutionary step of the internet of things," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, 2014.

[61] U. Sedlar, M. Rugelj, M. Volk, and J. Sterle, "Multi-layer end-to-end mobile network monitoring," in *ELMAR*, pp. 175–179, IEEE, 2015.

[62] R. Leira, G. Julián-Moreno, I. González, F. J. Gómez-Arribas, and J. E. L. de Vergara, "Performance assessment of 40 gbit/s off-the-shelf network cards for virtual network probes in 5g networks," *Computer Networks*, vol. 152, pp. 133–143, 2019.

[63] S. Khairi, B. Raouyane, and M. Bellafkih, "Novel qoe monitoring and management architecture with etom for sdn-based 5g networks," *Cluster Computing*, pp. 1–12, 2019.

[64] C. Tselios and G. Tsolis, "On qoe-awareness through virtualized probes in 5g networks," in *Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, pp. 159–164, IEEE, 2016.

[65] D. Adami, L. Donatini, G. Foddis, and S. e. Giordano, "Design and development of management functions for distributed monitoring based on sdn-based network," in *Euro Med Telco Conference (EMTC)*, pp. 1–5, IEEE, 2014.

[66] G. Bianchi, E. Biton, and N. e. Blefari-Melazzi, "Superfluidity: a flexible functional architecture for 5g networks," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1178–1186, 2016.

[67] T. Choi, S. Kang, S. Yoon, and S. e. Yang, "Suvmf: Software-defined unified virtual monitoring function for sdn-based large-scale networks," in *Future Internet Technologies*, pp. 4:1–4:6, ACM, 2014.

[68] Qosmos, "Layer 7 visibility for virtual cpe," 2016.

[69] Affirmed, "vprobe solution brief - affirmed networks," 2016.

[70] D. EMC, "Bringing virtual probes and analytics together for the next-generation mobile network," 2019.

[71] NetScout, "Netscout ngenius virtual agent," 2016.

[72] A. CORD, "Virtual probe," 2016.

[73] Sencore, "Virtual 10g monitoring probe," 2016.

[74] EXFO, "Active physical and software verifier," 2019.

[75] G. Americas, "4g americas - nfv and sdn network," 2015.

[76] H. Mahkonen, R. Manghirmalani, and M. e. Shirazipour, "Elastic network monitoring with virtual probes," in *Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 1–3, IEEE, 2015.

[77] Netrounds, "Ensuring end-user quality in nfv-based infrastructure," 2016.

[78] M. Xia, M. Shirazipour, and H. e. Mahkonen, "Resource optimization for service chain monitoring in software-defined networks," in *4th European Workshop on Software Defined Networks (EWSDN)*, pp. 91–96, IEEE, 2015.

[79] NICT, "Towards traceable overlay network over virtualized systems." `http://www.nict.go.jp/publication/shuppan/kihou-journal/journal-vol58no3_4/journal-vol58no3-4_0304.pdf`, 2011.

[80] J. Hunt, C. Hale, and J. Kappel, "End-to-end network service lifecycle demonstration," in *Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 9–12, IEEE, 2015.

[81] L. Dinh-Xuan, M. Seufert, F. Wamser, and P. Tran-Gia, "Study on the accuracy of qoe monitoring for http adaptive video streaming using vnf," in *1st IFIP/IEEE International Workshop on Quality of Experience Management (QoE-Management)*, 5 2017.

[82] Telechemy, "Managing performance in software defined and virtualized networks," 2014.

[83] Empirix, "Quality of experience in virtualized networks (nfv)," 2015.

[84] Qosmos, "Qosmos deepflow®: Quality of experience use case," 2013.

[85] Radcom, "A virtualized solution," 2017.

[86] Argella, "Performance and service quality monitoring on virtualized networks," 2018.

[87] I. Technologies, “Cellular virtual monitoring platform,” 2019.

[88] S. Les Cottrell, “Network monitoring tools,” 2017.

[89] E. Liotou, D. Tsolkas, and N. Passas, “A roadmap on qoe metrics and models,” in *Telecommunications (ICT), 2016 23rd International Conference on*, pp. 1–5, IEEE, 2016.

[90] sFlow, “sflow- making the network visible,” 2017.

[91] Cisco, “Cisco ios netflow - cisco,” 2012.

[92] Cisco, “Remote monitoring (rmon),” 1999.

[93] A. Claudio, R. Daniele, T. Christian, M. Christopher, and et. Al., “Automated qoe evaluation of dynamic adaptive streaming over http,” in *Quality of Multimedia Experience (QoMEX)*, pp. 58–63, IEEE, 2013.

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

[95] L. Skorin-Kapov, K. Ivesic, G. Aristomenopoulos, and S. Papavassiliou, “Approaches for utility-based qoe-driven optimization of network resource allocation for multimedia services,” in *Data traffic monitoring and analysis*, pp. 337–358, Springer, 2013.

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

[97] 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,” in *2012 14th International Conference on Advanced Communication Technology (ICACT)*, pp. 459–464, IEEE, 2012.

[98] H. J. Kim and et. Al., “Qoe assessment model for video streaming service using qos parameters in wired-wireless network,” in *International Advanced Communication Technology (ICACT)*, pp. 459–464, IEEE, 2012.

[99] “Xiph.org video test media [derf’s collection].”

[100] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards qoe-aware video streaming using sdn," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, pp. 1317–1322, IEEE, 2014.

[101] A. Bentaleb, A. C. Begen, and R. Zimmermann, "Sdndash: Improving qoe of http adaptive streaming using software defined networking," in *Proceedings of the 2016 ACM on Multimedia Conference*, pp. 1296–1305, ACM, 2016.

[102] R. V. Rosa, M. A. S. Santos, and C. E. Rothenberg, "Md2-nfv: The case for multi-domain distributed network functions virtualization," in *International Conference and Workshops on Networked Systems (NetSys)*, pp. 1–5, IEEE, 2015.

[103] B. Chatras and F. F. Ozog, "Network functions virtualization: the portability challenge," *IEEE Network*, vol. 30, no. 4, pp. 4–8, 2016.

[104] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[105] S. F. Fernandes, "Performance evaluation for network services, systems and protocols," 2017.

[106] K. Claffy and T. Monk, "What's next for internet data analysis? status and challenges facing the community," *Proceedings of the IEEE*, vol. 85, no. 10, pp. 1563–1571, 1997.

[107] M. C. Luizelli and L. R. e. Bays, "Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions," in *Integrated Network Management (IM)*, pp. 98–106, IEEE, 2015.

[108] U. Fiore, P. Zanetti, F. Palmieri, and F. Perla, "Traffic matrix estimation with software-defined nfv: Challenges and opportunities," *Journal of Computational Science*, 2017.

[109] M. Qiao, Y. Ma, Y. Bian, and J. Liu, "Real-time multi-application network traffic identification based on machine learning," in *International Symposium on Neural Networks*, pp. 473–480, Springer, 2015.

[110] W. De Donato, A. Pescapé, and A. Dainotti, "Traffic identification engine: an open platform for traffic classification," *IEEE Network*, vol. 28, no. 2, pp. 56–64, 2014.

[111] C. Hue, Y.-J. Chen, and L.-C. Wang, "Traffic-aware networking for video streaming service using sdn," in *Computing and Communications Conference (IPCCC)*, pp. 1–5, IEEE, 2015.

[112] Y. Wang, F. Liu, and Z. Lei, "Extracting and measuring quality of experience indicators of social business," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1, pp. 491–495, IEEE, 2013.

[113] A. Cuadra, M. Cutanda, A. Aurelius, and K. e. Brunnstrom, "Ecosystem for customer experience assurance," in *Smart Communications in Network Technologies (SaCoNeT)*, vol. 3, pp. 1–5, IEEE, 2013.

[114] E. I. Papagiannakopoulou, M. N. Koukovini, G. V. Lioudakis, and J. e. Garcia-Alfaro, "A privacy-aware access control model for distributed network monitoring," *Computers & Electrical Engineering*, vol. 39, no. 7, pp. 2263–2281, 2013.

[115] A. Antonakopoulou, F. Gogoulos, and G. e. Lioudakis, "An ontology for privacy-aware access control in network monitoring environments," *Journal of Research and Practice in Information Technology*, vol. 46, no. 4, 2014.

[116] L. Wei, H. Zhu, Z. Cao, and X. e. Dong, "Security and privacy for storage and computation in cloud computing," *Information Sciences*, vol. 258, pp. 371–386, 2014.

[117] L. Arockiam and S. Monikandan, "Efficient cloud storage confidentiality to ensure data security," in *Computer Communication and Informatics (ICCCI)*, pp. 1–5, IEEE, 2014.

[118] E. Liotou, D. Tsolkas, N. Passas, and L. Merakos, "Quality of experience management in mobile cellular networks: key issues and design challenges," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 145–153, 2015.

[119] V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, and P. Valduriez, "Streamcloud: A large scale data streaming system," in *Distributed Computing Systems (ICDCS)*, pp. 126–137, IEEE, 2010.

[120] D. Bonino and F. Corno, "spchains: A declarative framework for data stream processing in pervasive applications," *Procedia Computer Science*, vol. 10, pp. 316–323, 2012.

[121] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.

[122] M. Kutare, G. Eisenhauer, and C. e. Wang, "Monalytics: online monitoring and analytics for managing large scale data centers," in *Proceedings of the 7th international conference on Autonomic computing*, pp. 141–150, ACM, 2010.

[123] V. Duarte and N. Farruca, "Using libpcap for monitoring distributed applications," 2010.

[124] "What is a network protocol analyzer?."

[125] P. Asrodia and H. Patel, "Network traffic analysis using packet sniffer," 2012.

[126] ntop, "ntop," 2017.

[127] K. Roebuck, "Deep packet inspection: High-impact strategies - what you need to know," 2011.

[128] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[129] K. Phemius and M. Bouet, "Monitoring latency with openflow," in *9th International Conference on Network and Service Management (CNSM)*, pp. 122–125, IEEE, 2013.

[130] R. Mijumbi, J. Serrat, and J.-L. e. Gorricho, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[131] G. Dimopoulos, I. Leontiadis, and P. e. Barlet-Ros, "Identifying the root cause of video streaming issues on mobile devices," in *Emerging Networking Experiments and Technologies*, p. 24, ACM, 2015.

[132] M. Bouet, J. Leguay, T. Combe, and V. Conan, "Cost-based placement of vdpi functions in nfv infrastructures," *International Journal of Network Management*, vol. 25, no. 6, pp. 490–506, 2015.

[133] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Netw.*, vol. 25, no. 5, pp. 355–374, 2015.

[134] L. Deri, M. Martinelli, and A. Cardigliano, "Realtime high-speed network traffic monitoring using ntopng," in *28th Large Installation System Administration Conference (LISA14)*, pp. 78–88, 2014.

[135] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "ndpi: Open-source high-speed deep packet inspection," in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 617–622, IEEE, 2014.

[136] W. M. Shbair, T. Cholez, J. Francois, and I. Chrisment, "A multi-level framework to identify https services," in *Network Operations and Management Symposium (NOMS)*, pp. 240–248, IEEE, 2016.

[137] Allot, "Evolution of network service enablement utilizing nfv," 2016.

[138] A. Nourian and M. Maheswaran, "Privacy and security requirements of data intensive computing in clouds," in *Handbook of Data Intensive Computing*, pp. 501–518, Springer, 2011.

[139] D. Trihinas, G. Pallis, and M. D. Dikaiakos, "Jcatascopia: Monitoring elastically adaptive applications in the cloud," in *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 226–235, IEEE, 2014.

[140] V. Pilloni, P. Navaratnam, S. Vural, L. Atzori, and R. Tafazolli, "Tan: a distributed algorithm for dynamic task assignment in wsns," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1266–1279, 2014.

[141] J. Fink, "Docker: a software as a service, operating system-level virtualization framework," 2014.

[142] A. V. Romero, "Virtualbox 3.1: Beginner's guide," 2010.

[143] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: the linux virtual machine monitor," 2007.

[144] J. N. Matthews, E. M. Dow, and T. e. Deshane, "Running xen: a hands-on guide to the art of virtualization," 2008.

[145] E. Haletky, "Vmware esx and esxi in the enterprise: planning deployment of virtualization servers," 2011.

[146] C. Gormley and Z. Tong, "Elasticsearch: The definitive guide: A distributed real-time search and analytics engine," 2015.

[147] T. Grainger, T. Potter, and Y. Seeley, "Solr in action," 2014.

[148] I. Management Association, "Decision management: Concepts, methodologies, tools, and applications: Concepts, methodologies, tools, and applications," 2017.

[149] Joojip, "Joojip," 2016.

[150] huawei, "Building smart platform to assure qoe," 2011.

[151] C. Price, S. Rivera, *et al.*, "Opnfv: An open platform to accelerate nfv," *White Paper*, 2012.

[152] S. Crosby, R. Doyle, M. Gering, M. Gionfriddo, S. Grarup, S. Hand, M. Hapner, D. Hiltgen, *et al.*, "Open virtualization format specification," *vol. DSP0243*, vol. 1, no. 0, 2010.

[153] "Openstack enhanced platform awareness.."

[154] D. Lopez, "Openmano: The dataplane ready open source nfv mano stack," in *IETF Meeting Proceedings, Dallas, Texas, USA*, 2015.

[155] R. Mijumbi, J. Serrat, J.-l. Gorricho, and S. e. Latre, "Management and orchestration challenges in network functions virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, 2016.

[156] "Enhanced-platform-awareness-pcie."

[157] N. F. Virtualisation, "Nfv performance & portability best practises," *ETSI Standard GS NFV-PER*, vol. 1, 2014.

[158] ETSI, "Network functions virtualisation (nfv) release 3;security;security management and monitoring specification," 2017.

[159] M. D. Firoozjaei, J. P. Jeong, H. Ko, and H. Kim, "Security challenges with network functions virtualization," *Future Generation Computer Systems*, vol. 67, pp. 315–324, 2017.

[160] ETSI, "Network functions virtualisation (nfv); nfv security; security and trust guidance," *V1*, 2014.

[161] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[162] ETSI, "Network functions virtualisation - white paper 3," 2014.

[163] ETSI, "Network functions virtualisation (nfv);resiliency requirements," 2015.

[164] J. Martins, M. Ahmed, and C. e. Raiciu, "Clickos and the art of network function virtualization," in *USENIX Conference on Networked Systems Design and Implementation*, pp. 459–473, USENIX Association, 2014.

[165] X. Ge, Y. Liu, D. H. Du, and et.al., "Openanfv: Accelerating network function virtualization with a consolidated framework in openstack," in *ACM SIGCOMM Computer Communication Review*, no. 4, pp. 353–354, ACM, 2014.

[166] ETSI, "Network functions virtualisation, an introduction, benefits, enablers,challenges and call for action. issue 1," 2015.

[167] J. Batalle, J. F. Riera, E. Escalona, and J. A. Garcia-Espin, "On the implementation of nfv over an openflow infrastructure: Routing function virtualization," in *Future Networks and Services (SDN4FNS)*, pp. 1–6, IEEE, 2013.

[168] B. Rong, X. Qiu, M. Kadoch, and et.al., "Intelligent sdn and nfv for 5g hetnet dynamics," in *5G Heterogeneous Networks*, pp. 15–40, Springer, 2016.

[169] E. Liotou, H. Elshaer, and et. Al., "Shaping qoe in the 5g ecosystem," in *Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 2015.

[170] C. Tselios and G. Tsolis, "On qoe-awareness through virtualized probes in 5g networks," in *Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop*, pp. 159–164, IEEE, 2016.

[171] P. Trakas, F. Adelantado, and et. Al., "A quality of experience-aware association algorithm for 5g heterogeneous networks," in *Communications (ICC), 2017 IEEE International Conference*, pp. 1–6, IEEE, 2017.

[172] Y.-H. Chu, W.-T. Lin, and et. Al., "Software-defined qoe measurement architecture," in *Network Operations and Management Symposium (APNOMS)*, pp. 1–4, IEEE, 2014.

[173] E. Liotou, N. Passas, and L. Merakos, "Towards qoe provisioning in next generation cellular networks," *E-LETTER*, vol. 6, p. 2, 2015.

[174] A. Farshad, P. Georgopoulos, and et. Al., "Leveraging sdn to provide an in-network qoe measurement framework," in *Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 239–244, IEEE, 2015.

[175] S. Ramakrishnan, X. Zhu, and et. Al., "Sdn based qoe optimization for http-based adaptive video streaming," in *International Symposium on Multimedia (ISM)*, pp. 120–123, IEEE, 2015.

[176] O. Awobuluyi, J. Nightingale, and et. Al., "Video quality in 5g networks: Context-aware qoe management in the sdn control plane," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1657–1662, IEEE, 2015.

[177] L. Guillen, S. Izumi, T. Abe, and T. Suganuma, "Sand/3: Sdn-assisted novel qoe control method for dynamic adaptive streaming over http/3," *Electronics*, vol. 8, no. 8, p. 864, 2019.

[178] P. Velan, M. Čermák, and et. Al., "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[179] "Sdni: A message exchange protocol for software defined networks (sdns) across multiple domains."

[180] "Ipv6 flow label specification."

[181] "ipstack."

[182] M. Team, "Mininet."

[183] "Opendaylight."

[184] A. Najjar, X. Serpaggi, and et. Al., "Multi-agent systems for personalized qoe-management," in *International Teletraffic Congress (ITC 28)*, vol. 3, pp. 1–6, IEEE, 2016.

[185] X. Fang and e. A. Rui, Lanlan, "The selection of the monitored-set for qos collection based on device-agent," in *International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pp. 1–5, IEEE, 2010.

[186] M. Alreshoodi and J. Woods, "Survey on qoe\qos correlation models for multimedia services," *arXiv preprint arXiv:1306.0221*, 2013.

[187] E. Liotou, A. Marotta, and et. Al., "A middleware architecture for qoe provisioning in mobile networks," in *International Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–5, IEEE, 2017.

[188] M. Siller and J. Woods, "Using an agent based platform to map quality of service to experience in conventional and active networks," *IEE Proceedings-Communications*, vol. 153, no. 6, pp. 828–840, 2006.

[189] S. Y. Yoon, S. Lee, and et. Al., "Mobile data service qoe analytics and optimization," in *IEEE International Conference on Communication Workshop (ICCW)*, pp. 1699–1704, IEEE, 2015.

[190] L. Zhang, S. Wang, and et. Al., "Qoecenter: A visual platform for qoe evaluation of streaming video services," in *International Conference on Web Services (ICWS)*, pp. 212–219, IEEE, 2017.

[191] G. Gómez, J. Lorca, and et. Al., "Towards a qoe-driven resource control in lte and lte-a networks," *Journal of Computer Networks and Communications*, vol. 2013, 2013.

[192] P. L. Mateo Navarro and et. Al., "A context-aware interaction model for the analysis of users' qoe in mobile environments," *International Journal of Human-Computer Interaction*, vol. 30, no. 12, pp. 946–964, 2014.

[193] E. Liotou, G. Tseliou, K. Samdanis, and et. Al., "An sdn qoe-service for dynamically enhancing the performance of ott applications," in *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*, pp. 1–2, IEEE, 2015.

[194] A. J. Verdejo, K. De Moor, and et. Al., "Qoe estimation of a location-based mobile game using on-body sensors and qos-related data," in *Wireless Days (WD), 2010 IFIP*, pp. 1–5, IEEE, 2010.

[195] A. Ahmad, A. Floris, and L. Atzori, "Ott-isp joint service management: a customer lifetime value based approach," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 1017–1022, IEEE, 2017.

[196] Y. Wang, W. Zhou, and P. Zhang, *QoE Management in Wireless Networks*. Springer, 2017.

[197] A. Ahmad, A. Floris, and L. Atzori, "Towards qoe monitoring at user terminal: A monitoring approach based on quality degradation," in *International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, IEEE, 2017.

[198] P. Reichl, "From charging for quality of service to charging for quality of experience," *annals of telecommunications-annales des télécommunications*, vol. 65, no. 3-4, pp. 189–199, 2010.

[199] E. Grigoriou, A. A. Barakabitze, L. Atzori, L. Sun, and V. Pilloni, "An sdn-approach for qoe management of multimedia services using resource allocation," in *IEEE International Conference on Communications 2017*, pp. 1–7, IEEE, 2017.

[200] "Iperf: Tcp/udp bandwidth measurement tool."

[201] "qperf - measure rdma and ip performance."

[202] A. Ahmad, A. Floris, and L. Atzori, "Qoe-aware service delivery: a joint-venture approach for content and network providers," in *International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 2016.

[203] S. S. Cherian and A. N. Rudrapatna, "Lte location technologies and delivery solutions," *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 175–194, 2013.

[204] E. Grigoriou, T. Saoulidis, L. Atzori, and et.Al., "An agent-based qoe monitoring strategy for lte networks," in *International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.

[205] "Elasticsearch-kibana-logstash."

[206] M. D. S. Deulkar and R. Deshmukh, “Data mining classification,” *Imperial Journal of Interdisciplinary Research*, vol. 2, no. 4, 2016.

[207] A. Imran, A. Zoha, and A. Abu-Dayya, “Challenges in 5g: how to empower son with big data for enabling 5g,” *IEEE network*, vol. 28, no. 6, pp. 27–33, 2014.

[208] S. Baraković and L. Skorin-Kapov, “Survey and challenges of qoe management issues in wireless networks,” *Journal of Computer Networks and Communications*, vol. 2013, 2013.

[209] E. Grigoriou, T. Saoulidis, L. Atzori, V. Pilloni, and P. Chatzimisios, “An agent-based qoe monitoring strategy for lte networks,” in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.

[210] A. Zabrovskiy, E. Kuzmin, E. Petrov, and M. Fomichev, “Emulation of dynamic adaptive streaming over http with mininet,” in *Proceedings of the 18th Conference of Open Innovations Association FRUCT*, pp. 391–396, FRUCT Oy, 2016.

[211] K. Piamrat, C. Viho, J.-M. Bonnin, and A. Ksentini, “Quality of experience measurements for video streaming over wireless networks,” in *Information Technology: New Generations, 2009. ITNG’09. Sixth International Conference on*, pp. 1184–1189, IEEE, 2009.

[212] V. Tsibonis, L. Georgiadis, and L. Tassiulas, “Exploiting wireless channel state information for throughput maximization,” in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 1, pp. 301–310, IEEE, 2003.

[213] M. Shehada, S. Thakolsri, Z. Despotovic, and W. Kellerer, “Qoe-based cross-layer optimization for video delivery in long term evolution mobile networks,” in *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 1–5, IEEE, 2011.

[214] F. Kelly, “Charging and rate control for elastic traffic,” *European transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.

[215] R. M. Abuteir, A. Fladenmuller, and O. Fourmaux, “Sdn based architecture to improve video streaming in home networks,” in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 220–226, IEEE, 2016.

[216] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, “Towards qoe-aware video streaming using sdn,” in *2014 IEEE Global Communications Conference*, pp. 1317–1322, IEEE, 2014.

[217] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, O. L. Pérez, M. U. Itzazelaia, I. Ahmad, M. Liyanage, *et al.*, "Sdn and nfv integration in generalized mobile network architecture," in *2015 European conference on networks and communications (EuCNC)*, pp. 154–158, IEEE, 2015.

[218] I. Ahmad, M. Liyanage, S. Namal, M. Ylianttila, A. Gurtov, M. Eckert, T. Bauschert, Z. Faigl, L. Bokor, E. Saygun, *et al.*, "New concepts for traffic, resource and mobility management in software-defined mobile networks," in *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pp. 1–8, IEEE, 2016.

[219] H. Koumaras, C. Sakkas, M. A. Kourtis, C. Xilouris, V. Koumaras, and G. Gardikis, "Enabling agile video transcoding over sdn/nfv-enabled networks," in *2016 International Conference on Telecommunications and Multimedia (TEMU)*, pp. 1–5, IEEE, 2016.

[220] L. De Cicco, S. Mascolo, and V. Palmisano, "Qoe-driven resource allocation for massive video distribution," *Ad Hoc Networks*, vol. 89, pp. 170–176, 2019.

[221] B. Dudin, N. A. Ali, A. Radwan, and A.-E. M. Taha, "Resource allocation with automated qoe assessment in 5g/b5g wireless systems," *IEEE Network*, vol. 33, no. 4, pp. 76–81, 2019.

[222] A. Kassler, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, and P. Dely, "Towards qoe-driven multimedia service negotiation and path optimization with software defined networking," in *SoftCOM 2012, 20th International Conference on Software, Telecommunications and Computer Networks*, pp. 1–5, IEEE, 2012.

[223] T. Hoßfeld, D. Hausheer, F. V. Hecht, F. Lehrieder, S. Oechsner, I. Papafili, P. Racz, S. Soursos, D. Staehle, G. D. Stamoulis, *et al.*, "An economic traffic management approach to enable the triplewin for users, isps, and overlay providers.," in *Future Internet Assembly*, pp. 24–34, 2009.

[224] L. Skorin-Kapov and M. Matijasevic, "Modeling of a qos matching and optimization function for multimedia services in the ngn," in *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, pp. 55–68, Springer, 2009.

[225] VideoLAN, "Vlc media player."

[226] S. Hemminger *et al.*, "Network emulation with netem," in *Linux conf au*, pp. 18–23, 2005.

[227] 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, 2016.

[228] L. Dinh-Xuan, M. Seufert, F. Wamser, and P. Tran-Gia, "Study on the accuracy of qoe monitoring for http adaptive video streaming using vnf," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, pp. 999–1004, IEEE, 2017.

[229] T. HoBfeld, M. Varela, P. E. Heegaard, and L. Skorin-Kapov, "Observations on emerging aspects in qoe modeling and their impact on qoe management," in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 2018.

[230] M. G. Martini, C. Hewage, M. M. Nasrall, and O. Ognenoski, "Qoe control, monitoring, and management strategies," *Multimedia Quality of Experience (QoE): Current status and future requirements*, pp. 149–167, 2016.

[231] D. Dong and J. Herbert, "Energy efficient vm placement supported by data analytic service," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pp. 648–655, IEEE, 2013.

[232] I. Giannoulakis, J. O. Fajardo, J. G. Lloreda, P. S. Khodashenas, C. Ruiz, A. Betzler, E. Kafetzakis, J. Pérez-Romero, A. Albanese, M. Paolino, *et al.*, "Enabling technologies and benefits of multi-tenant multi-service 5g small cells," in *Networks and Communications (EuCNC), 2016 European Conference on*, pp. 42–46, IEEE, 2016.

[233] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pp. 1346–1354, IEEE, 2015.

[234] N. Bouten, J. Famaey, R. Mijumbi, B. Naudts, J. Serrat, S. Latré, and F. De Turck, "Towards nfv-based multimedia delivery," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pp. 738–741, IEEE, 2015.

[235] F. Malandrino, C. Casetti, G. Landi, *et al.*, "Optimization-in-the-loop for energy-efficient 5g," in *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pp. 1–9, IEEE, 2018.

[236] F. Wang, R. Ling, J. Zhu, and D. Li, "Bandwidth guaranteed virtual network function placement and scaling in datacenter networks," in *2015 IEEE 34th*

International Performance Computing and Communications Conference (IPCCC), pp. 1–8, IEEE, 2015.

[237] L. Gupta, M. Samaka, R. Jain, A. Erbad, D. Bhamare, and C. Metz, "Colap: A predictive framework for service function chain placement in a multi-cloud environment," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–9, IEEE, 2017.

[238] F. B. Jemaa, G. Pujolle, and M. Pariente, "Qos-aware vnf placement optimization in edge-central carrier cloud architecture," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, 2016.

[239] P.-W. Chi, Y.-C. Huang, and C.-L. Lei, "Efficient nfv deployment in data center networks," in *2015 IEEE International Conference on Communications (ICC)*, pp. 5290–5295, IEEE, 2015.

[240] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 725–739, 2016.

[241] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann, and W. Kellerer, "Towards a cost optimal design for a 5g mobile core network based on sdn and nfv," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 1061–1075, 2017.

[242] T.-M. Nguyen, S. Fdida, and T.-M. Pham, "A comprehensive resource management and placement for network function virtualization," in *2017 IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–9, IEEE, 2017.

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

[244] B. Zhang, J. Hwang, and T. Wood, "Toward online virtual network function placement in software defined networks," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pp. 1–6, IEEE, 2016.

[245] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "Qos-driven function placement reducing expenditures in nfv deployments," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2017.

[246] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 3, pp. 1818–1831, 2017.

[247] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards qos-aware fog service placement," in *2017 IEEE 1st international conference on Fog and Edge Computing (ICFEC)*, pp. 89–96, IEEE, 2017.

[248] R. Cziva and D. P. Pezaros, "On the latency benefits of edge nfv," in *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 105–106, IEEE, 2017.

[249] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient nfv-enabled mobile edge-cloud for low latency mobile applications," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 475–488, 2018.

[250] P. Minoves, O. Frendved, B. Peng, A. Mackarel, and D. Wilson, "Virtual cpe: Enhancing cpe's deployment and operations through virtualization," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pp. 687–692, IEEE, 2012.

[251] M. Team, "Mininet," 2014.

[252] C. Mueller, S. Lederer, J. Poecher, and C. Timmerer, "Demo paper: Libdash-an open source software library for the mpeg-dash standard," in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pp. 1–2, IEEE, 2013.

[253] H. Balwani, P. Wagh, R. Vadaje, S. Shivgunde, and M. Wakode, "‘implementation of qoe/qos mapping in sdn," *Int. Res. J. Eng. Technol*, vol. 4, no. 4, pp. 957–959, 2017.

[254] F. Benamrane, M. B. Mamoun, and R. Benaini, "Performances of openflow-based software-defined networks: an overview," *Journal of Networks*, vol. 10, no. 6, pp. 329–338, 2015.

[255] E. Miravalls-Sierra, D. Muelas, J. López de Vergara, J. Ramos, and J. Aracil, "On the use of affordable cots hardware for network measurements: Limits and good practices," *Information*, vol. 9, no. 2, p. 43, 2018.

[256] W. S. Lima, E. Souto, T. Rocha, R. W. Pazzi, and F. Pramudianto, "User activity recognition for energy saving in smart home environment," in *2015 IEEE Symposium on Computers and Communication (ISCC)*, pp. 751–757, IEEE, 2015.

[257] V. Pilloni, A. Floris, A. Meloni, and L. Atzori, “Smart home energy management including renewable sources: A qoe-driven approach,” *IEEE Transactions on Smart Grid*, 2016.

[258] S. Chen, T. Liu, F. Gao, J. Ji, Z. Xu, B. Qian, H. Wu, and X. Guan, “Butler, not servant: A human-centric smart home energy management system,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 27–33, 2017.

[259] H. Lee, W.-K. Park, and I.-W. Lee, “A home energy management system for energy-efficient smart homes,” in *Computational Science and Computational Intelligence (CSCI), 2014 International Conference on*, vol. 2, pp. 142–145, IEEE, 2014.