TÉCNICO LISBOA

Provision of Multiuser 360° Video Streaming over MEC-enabled 5G Networks

Miguel Sampaio Crena Loff Barreto

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. António José Castelo Branco Rodrigues
Prof. Ivo Luís de la Cerda Garcia e Sousa

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. António José Castelo Branco Rodrigues
Member of the Committee: Prof. Francisco António Bucho Cercas

November 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First and foremost, I would like to express my sincere thanks to my supervisors, Prof. António Rodrigues, Prof. Maria Paula Queluz and Prof. Ivo Sousa. Their helpful advice, useful insights, availability and patience have proved crucial toward the successful completion of this work. The knowledge Professors Rodrigues, Queluz and Sousa have shared with me has allowed me to develop a comprehensive understanding of this dynamic and fast-growing area of technology and also plenty of other skills which I am sure will be key to my future success.

My thanks also go to Instituto de Telecomunicações (IT) for providing me with a workspace with the required means for running the simulations. Especially, I would like to thank Paulo Barata for his availability, professionalism and technical support.

To all my friends who accompanied me throughout this journey, I would not have done it without them. Thank you Miguel Marques, Maria Malaquias, Maria Inês Mendes, Inês Paulino, Tomás Ferreira Duarte, Ricardo Oliveira, António Costa, Margarida Figueira, Bernardo Santos, Gonçalo Nilo Fonseca, Malak Fawzy and many others, for all the encouragement, support, advice and much needed distractions. To everyone at Associação dos Estudantes do Instituto Superior Técnico (AEIST), where I had the pleasure of spending the better part of the last few years, thank you for giving me the space and opportunity to profoundly grow and develop myself as a person and as a professional.

Finally, I would like to express my deepest gratitude to my family for their unconditional and unfailing support and for always believing in me. To my mother and father, Maria João and João, for their hard work to provide me with every possible means to reach this milestone. To my sisters, Carolina and Ana, my brother-in-law, Emanuel, my nephews, João, Afonso and Tomás, and my grandparents for teaching me the value of family.

This work was funded by FCT/MCTES through national funds, and when applicable co-funded by EU funds, under the research grant UID/EEA/50008/2020.

Abstract

Data traffic on the next generation of wireless networks (5G) is expected to be dominated by challenging video applications, such as 360º video streaming. In order for 5G networks to handle this and other demanding applications while meeting the users' Quality of Experience (QoE) requirements, it is necessary to characterize the network side factors, such as the deployment of Multi-Access Edge Computing (MEC) infrastructure and the used video delivery scheme, jointly with the 360º video features.

To this end, the purpose of this thesis is to assess and compare the performance of different multiuser 360º video streaming solutions proposed in the literature. Since the high bandwidth requirements of 360º video delivery strongly limit the number of supported simultaneous streaming sessions, where users are watching different videos with different viewports, it is important to optimize the resource allocation in order to maximize the number of users that can satisfactorily stream 360º videos. Furthermore, this thesis aims at assessing the performance gains that can be achieved by the deployment of MEC infrastructure in the network.

It is found that using a tiles-based streaming scheme increases the number of very satisfied users by 50% over the traditional viewport-independent scheme. Furthermore, it is found that deploying MEC servers on the edge of the network unlocks the possibility of using a viewport- rendering scheme, which allows a further three-fold capacity gain over the tiles-based scheme, while virtually eliminating non-satisfied users.

Keywords: 360° video streaming, viewport adaptive streaming, 5G, QoE, MPEG-DASH

Resumo

Prevê-se que o tráfego de dados nas redes sem-fios de próxima geração (5G) seja dominado por aplicações intensivas de vídeo, como o *streaming* de vídeo 360º. De forma a permitir que as redes 5G suportem esta e outras aplicações mantendo os requisitos de Qualidade de Experiência (QoE) dos utilizadores, é necessário caracterizar os fatores relativos à rede, como a existência de infraestrutura Multi-Access Edge Computing (MEC) e o esquema de entrega de vídeo, em conjunto com as particularidades do vídeo 360º.

Neste sentido, o objetivo desta tese é avaliar e comparar o rendimento de diferentes soluções de *streaming* multi-utilizador de vídeo 360º propostas na literatura. Uma vez que os elevados requisitos de largura de banda dos esquemas de entrega de vídeo 360º limitam fortemente o número de sessões de streaming simultâneas suportadas, é importante otimizar a alocação de recursos de forma a maximizar este número. Para além disto, esta tese tem como objetivo avaliar os ganhos de rendimento que conseguem ser atingidos ao utilizar infraestrutura MEC na rede.

Conclui-se que utilizar um esquema de *streaming* baseado em *tiles* aumenta em 50% o número de utilizadores muito satisfeitos relativamente ao esquema tradicional independente do *viewport* do utilizador. Para além disso, conclui-se que a utilização de servidores MEC na orla da rede desbloqueia a possibilidade de utilizar um esquema de *streaming* baseado em pré-processamento dos *viewports*, o que permite um ganho de capacidade de três vezes em relação ao esquema de *tiles*, virtualmente eliminando ao mesmo tempo os utilizadores não-satisfeitos.

Palavras-chave: *streaming* de vídeo 360°, *streaming* adaptativo ao *viewport*, 5G, QoE, MPEG-DASH

Contents

List of Tables

List of Figures

List of Abbreviations

3GPP	Third Generation Partnership Project
4G	4^{th} Generation
5G	5^{th} Generation
5GC	5G Core
ABR	Adaptive Bitrate
AMC	Adaptive Modulation and Coding
AMF	Access and Mobility Management Function
AR	Augmented Reality
AVC	Advanced Video Coding
BBU	Baseband Unit
BD-PSNR	Bjøntegaard Delta PSNR
BLER	Block Error Rate
BS	Base Station
C-RAN	Cloud RAN
CA	Carrier Aggregation
CAGR	Compound Annual Growth Rate
CAPEX	Capital Expenditures
CC	Component Carrier
CMP	Cubemap Projection
CP-OFDM	Cyclic Prefix Orthogonal Frequency Division Multiplexing
CQI	Channel Quality Indicator
CU	Central Unit
DFT-s-OFDM	Discrete Fourier Transform Spread OFDM
DMOS	Difference Mean Opinion Score
DU	Distributed Unit
eMBB	Enhanced Mobile Broadband
EPC	Evolved Packet Core
ERP	Equirectangular Projection
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FoV	Field of View
FR1	Frequency Region 1

FR2	Frequency Region 2
gNB	Next Generation NodeB
GOP	Group of Pictures
HEVC	High-Efficiency Video Coding
HMD	Head-Mounted Display
HTTP	Hypertext Transfer Protocol
HVS	Human Vision System
IMT-2020	International Mobile Telecommunications-2020
IoT	Internet of Things
ITU-R	Radiocommunication Sector of the International Telecommunication Union
ITU-T	Telecommunication Sector of International Telecommunication Union
LDPP	Low Delay P Profile
LTE	Long Term Evolution
MCS	Modulation Coding Scheme
MEC	Multi-Acess Edge Computing
MIMO	Multiple Input Multiple Output
mMTC	Massive Machine Type Communications
MOS	Mean Opinion Score
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MPEG-DASH	MPEG Dynamic Adaptive Streaming over HTTP
MSE	Mean Squared Error
NFV	Network Function Virtualization
NG-RAN	Next-Generation Radio Access Network
NR	New Radio
NSA	Non-standalone
OCM	Offset Cubemap
OFDM	Orthogonal Frequency Division Multiplexing
OMAF	Omnidirectional MediA Format
OPEX	Operational Expenditures
OTT	Over the Top
PAPR	Peak-to-Average Power Ratio
PF	Proportional Fair
PRB	Physical Resource Block
PSNR	Peak-Signal-to-Noise Ratio
QAAD	QoE-enhanced Adaptation Algorithm over DASH
QAM	Quadrature Amplitude Modulation

QER	Quality Emphasis Region
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameter
QPSK	Quadrature Phase-Shift Keying
RAN	Radio Access Network
RAP	Random Access Profile
RAT	Radio Access Technology
RD	Rate-Distortion
RE	Resource Element
RTMP	Real-time Messaging Protocol
RTT	Round Trip Time
RU	Remote Radio Unit
SA	Standalone
SCS	Subcarrier Spacing
SDN	Software Defined Networking
SNR	Signal-to-noise Ratio
SRD	Spatial Representation Description
TDD	Time Division Duplex
TTI	Time Transmission Interval
UE	User Equipment
UPF	User Plane Function
URLLC	Ultra-reliable and Low Latency Communications
V-PSNR	Viewport PSNR
VAS	Viewport Adaptive Streaming
VR	Virtual Reality

Chapter 1

Introduction

This chapter provides the scope of the work developed in the context of this thesis. The motivation for solving the presented problem is presented, as well as the objectives that are expected to be achieved. The main contributions of this thesis and its structure are also described.

1.1 Motivation

Roughly every ten years, there is a revolution in mobile telecommunications technology. With each of these revolutions, the paradigm of what is possible to achieve with a wireless network shifts, unlocking completely new use cases. With the 1st Generation came the first analog cellular service, allowing users to make voice calls. The 2nd Generation introduced the use of digital signals, which unlocked a more efficient use of spectrum and the possibility to send data in addition to voice calls, creating a new use case for text messages. The 3rd Generation brought a profound revolution with much faster data rates, which, coupled with the introduction of smartphones, transformed the way users regard their mobile devices, using them to access e-mail, browse the Internet, and send and receive multimedia content. The current generation, 4th Generation (4G), introduced in 2008, adopted packet-switched networks for all communications. 4G has further expanded the pervasiveness of mobile devices and their capabilities, with high data rates which allow users to stream music and videos with high quality, make live streams from their mobile devices, etc.

The next generation of wireless communications systems, the 5th Generation (5G), aims at keeping up with the ever-growing global mobile data traffic, while delivering a leap forward in data rates, latency, connectivity, and more. It is forecast that, in 2026, there will be 3.5 billion 5G subscriptions, accounting for 40% of all mobile service subscriptions, and that 5G data traffic will account for 122 exabytes per month [1]. The forecast growth in data traffic until 2026 is illustrated in Figure 1.1.

Currently, video traffic accounts for 66% of all mobile data traffic, a figure that is expected to increase to 77% by 2026 [1]. This growth is going to be driven by challenging new use cases unlocked by the 5G networks, such as mobile broadcasting, remote surgery and Augmented Reality (AR).

Virtual Reality (VR) and AR applications are particularly interesting use cases for 5G networks, having the potential to emerge as one of the early opportunities for mobile network operators. It is identified that this market will be worth $292 billion by 2025 [2]. Globally, AR and VR traffic will grow nearly 12 times from 22 petabytes per month in 2017 to 254 petabytes in 2022 [3].

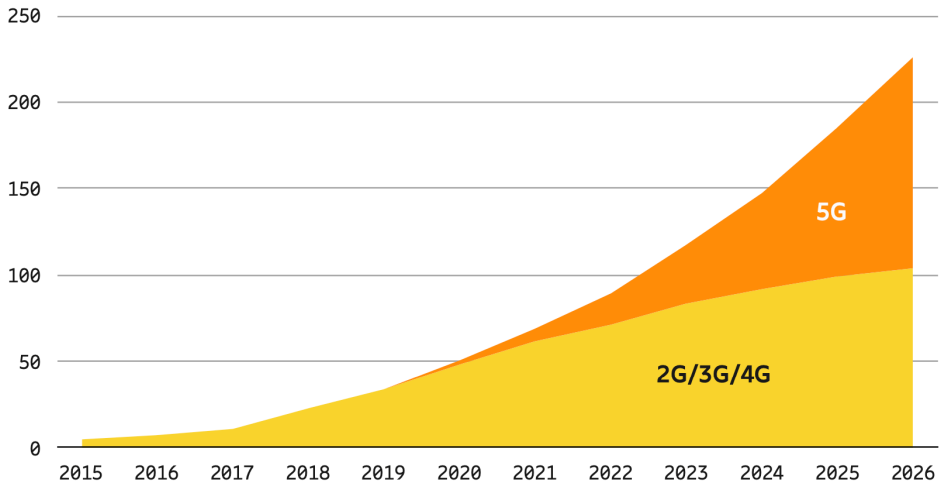


Figure 1.1: Global mobile data traffic forecast (exabytes/month) [1].

One of the most well developed and adopted VR/AR applications is 360° video, and therefore it is likely that this application will be one of the first ones to gain popularity. This is an application where users watch videos such as sports, live shows, documentaries and more, with the freedom to look around the entire 360° sphere around the point of view of the camera. In order to do this, they use a Head-Mounted Display (HMD) device, which allows them to look around, with the displayed video accompanying the head motion. The current availability of omnidirectional cameras make it easy to produce personalized 360° videos and social media platforms such as Facebook and YouTube already allow users to publish these videos [4].

Currently, this application poses significant challenges for wireless networks since it requires the transfer and storage of large amounts of data and demand significant computing resources. HMD devices have limited communication and storage resources and finite processing capabilities. Therefore, this and other VR/AR applications are dependent on the rollout of 5G networks, which, with their more ambitious data rate and latency targets, will enable new schemes for the delivery of the content and expand the capabilities of the network to support it.

However, it is still necessary to develop these schemes in order to cope with the ultra-high data rate and ultra-low latency requirements of 360° video in order to be able to support more than a few simultaneous users and therefore justify the investment in infrastructure. It is also necessary to study to which extent do the improvements enabled by 5G influence the users' Quality of Experience (QoE).

1.2 Objectives

One of the current challenges for mobile networks is to provide quality of experience for new and upcoming multimedia applications. Several delivery schemes have been proposed in the literature with the aim of reducing the necessary bandwidth necessary for 360° video streaming and improving its end-to-end latency, therefore improving the experience for users.

The main objective of this thesis is to study how these schemes compare against each other and to study in which ways the use of 5G networks improves the streaming experience for users. It also studied how one of the most promising new technologies of 5G, Multi-Access Edge Computing (MEC), may be beneficial for these applications.

1.3 Main Contributions

The main contribution of this thesis is the comparison of different 360° video delivery schemes in terms of the users' QoE. Through the development of a simulator, it is found that using a tiles-based streaming scheme increases the number of very satisfied users by 50% over the traditional viewport-independent scheme. Furthermore, it is found that deploying MEC servers in the edge of the network unlocks the possibility of using a viewport-rendering scheme, which allows a further three-fold gain in capacity over the tiles-based scheme, while eliminating non-satisfied users.

1.4 Thesis Outline

This thesis is organized in six chapters.

Chapter 1 introduces the motivations for the presented work, describing its objectives and main contributions.

Chapter 2 provides an overview of the new 5G standard, describing its key aspects, technologies and architecture. It is aimed at providing all the necessary concepts to understand how multimedia streaming works within a wireless communications environment.

Chapter 3 describes the state-of-the-art in 360° video streaming. It analyzes the challenges involved and presents a framework for creating and distributing 360° videos, including how it is transmitted and the various possible strategies to do this. It also explains why 5G is a pivotal enabler for this use case and introduces the concept of QoE, which will be later used to measure the users' satisfaction with the application. Finally, some state-of-the art approaches for this application are described.

Chapter 4 describes, in detail, the implementation of the simulator developed to assess the performance of the 5G network for this application, and the various delivery schemes.

Chapter 5 presents the assessment methodology and describes the conditions in which the simulations were made. It then presents and discusses the main results from these assessments.

Chapter 6 presents the conclusions of this thesis and provides some suggestions for future work.

Chapter 2

5G Overview

This chapter outlines the key aspects of 5G. An overview of the 5G use cases and capabilities is presented in Section 2.1. Section 2.2 presents the key technologies of 5G, while Section 2.3 gives an overview of the architecture. Section 2.4 describes the physical layer. Section 2.5 addresses 5G performance metrics. Finally, Section 2.6 presents the Multi-Acess Edge Computing technology.

2.1 Introduction

In this section, an overview of the 5G improvements, use cases and capabilities is given, based on the literature in [1, 5–9].

5G is the new standard of mobile access technology. Following 4G Long Term Evolution (LTE), it aims to be a leap forward in terms of data rates, latency, reliability, capacity and availability. 5G represents a major step in the capabilities that are expected from mobile networks, as it will enable services like Internet of Things (IoT) on a massive scale, data rates of up to 20 Gbps and a capacity increase of up to 1000 times.

5G networks enable new use cases for consumers, enterprises, homes, and public domains. The 360° video streaming application pertains to the consumer segment, which will be enhanced by the available higher data rate and lower latency.

With 5G enabling a much higher capacity when compared to LTE, it is expected that the global Internet traffic increases exponentially. By the end of 2019, the total global mobile traffic reached around 33 exabytes per month. Leveraged by 5G networks, this figure is expected to reach 226 exabytes per month in 2026, a time by which 54% of total mobile data will be generated by 5G networks.

The requirements of 5G are driven by the International Mobile Telecommunications-2020 (IMT-2020) standard and are issued by the Radiocommunication Sector of the International Telecommunication Union (ITU-R). A comparison between the requirements of IMT-2020 and the requirements of IMT-Advanced, which defines 4G, can be seen in Figure 2.1. There is a tenfold increase in the user experienced data rate, from 10 to 100 Mbps, as well as a ten-time reduction in latency, from 10 ms to 1 ms. Other key performance requirements of IMT-2020 are a download peak data rate of 20 Gbps, a threefold increase in spectrum efficiency, support for device mobility up to 500 km/h, a low energy mode when there is no data to transmit and a connection density of 1,000,000 devices/km^2.

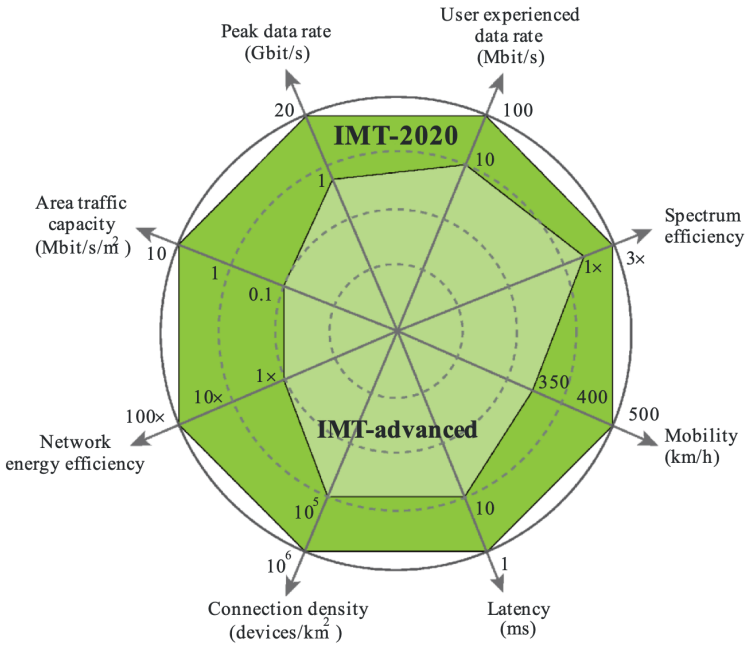


Figure 2.1: Comparison between IMT-2020 and IMT-Advanced requirements [6].

5G Usage Scenarios

IMT-2020 is aimed at supporting a wide range of use cases in three broad usage scenarios: Enhanced Mobile Broadband (eMBB), Ultra-reliable and Low Latency Communications (URLLC) and Massive Machine Type Communications (mMTC). Figure 2.2 depicts several possible 5G use cases and where they fit within the usage scenarios.

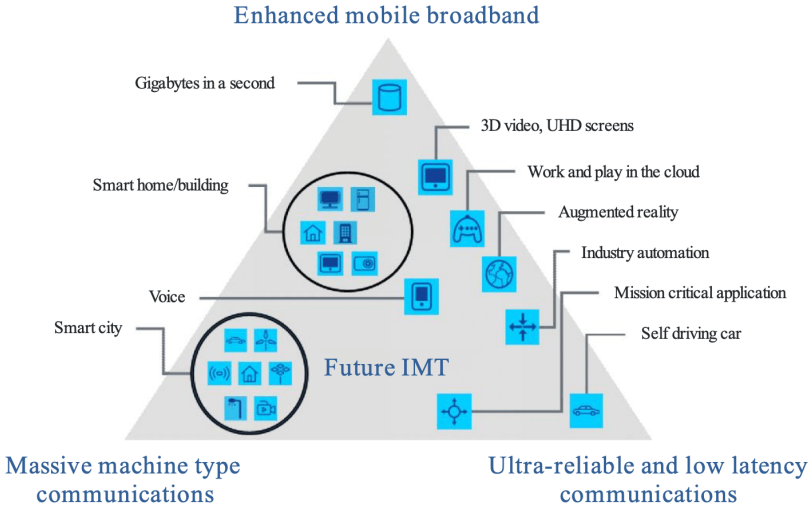


Figure 2.2: 5G use cases [6].

- **eMBB** - Addresses the human-centric use cases for access to multimedia content, services and data. It supports improved performance and an increasingly seamless usage experience compared to existing mobile broadband applications. This usage scenario covers use cases such as wide-area coverage and hotspots, enhancing broadband access in densely populated areas

and boosting indoor and outdoor coverage. The required peak download data rate is as high as 20 Gbps, and the maximum supported speed of the User Equipment (UE) is 500 km/h;

- **URLLC** - Addresses stringent requirements for capabilities such as throughput, latency and availability for delay-sensitive applications such as intelligent transport systems, vehicle-to-everything, remote medical surgery, smart grids, among others. The URLLC usage scenario user plane latency requirement is 1 ms;
- **mMTC** - Family of applications that consist of a very large number of devices typically transmitting a relatively low volume of non-delay-sensitive data. These applications may be IoT or Machine-to-Machine. IoT can be defined as intelligently connected small devices and sensors in machines and everyday objects exchanging data between each other. This usage scenario is also characterized by having a very high requirement for supporting connection density, as high as 1,000,000 devices/km^2.

2.2 5G Key Technologies

The demanding requirements defined by IMT-2020 require new technologies and advancements in existing ones. This section presents the key technologies of 5G, based on [5, 7–9]:

- **New spectrum** - 5G is the first mobile radio technology designed to operate on any frequency band between 400 MHz and 90 GHz. This wide range is required both for providing wide coverage with the low bands and high data rates and capacity with the high bands. The sub-6 GHz spectrum is also heavily congested, limiting the available bandwidth. In the centimeter and millimeter wave bands, bandwidths of the order of 1 GHz are available;
- **Beamforming** - Multiple Input Multiple Output (MIMO) antenna arrays at the base station will increase spectral efficiency and network coverage, especially at higher frequencies. Typical antenna numbers for the base station will vary between 256 to 1024 for the millimeter wave bands. Massive MIMO networks will utilize beamforming technology, enabling the targeted use of spectrum;
- **Network slicing** - Network slicing will create virtual network segments for the different services within the same physical 5G network. This is important because 5G needs flexible design to support all the different use cases, without the need for operators to build dedicated networks. Network slicing constructs different logical networks on a unified physical infrastructure via Software Defined Networking (SDN) and Network Function Virtualization (NFV);
- **Multi-Acess Edge Computing (MEC)** - The low latency requirements of 5G requires the content to be brought close to the radio, which will be enabled by edge computing. MEC can optimize mobile resources by hosting compute-intensive applications, process large data before sending to the cloud, provide cloud-computing capabilities within the Radio Access Network (RAN) in close proximity to mobile users, and offer context-aware services with the help of RAN information. MEC will enable a wide variety of applications where low latency is required, such as driverless vehicles, augmented reality and 360° video streaming;

2.3 5G Architecture

In order to guarantee the fulfilment of the expected requirements of extreme bandwidth and low latency in 5G networks, there is a need to develop a new architecture for the network. This new architecture will not only come with a new core network, 5G Core (5GC), but also with a new Radio Access Technology (RAT), called 5G New Radio (NR). 5G NR is part of the Release 15 specification from the Third Generation Partnership Project (3GPP) [10].

This section goes over the two major 5G architecture options, as well as an overview of the core network architecture and the RAN architecture. The information provided in this section is based on the literature in [5, 7, 10–19].

2.3.1 5G Architecture Options

There are several different factors that need to be considered when deciding the 5G network architecture. One of these factors is the need to interwork with 4G LTE, especially in the early phases of deployment. Therefore, as can be seen in Figure 2.3, there are two main architecture options:

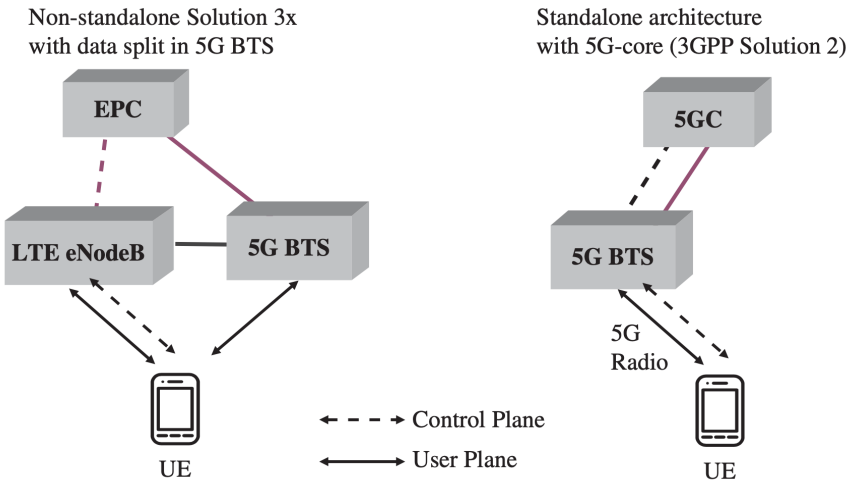


Figure 2.3: Main architecture options for 5G [5].

- **Non-standalone (NSA) 5G NR** - NSA refers to the architecture option where the existing LTE radio access and core network, Evolved Packet Core (EPC), are used as an anchor for 5G NR using dual connectivity. This means that UEs can use the radio resources provided by the two different network access points. This is the architecture that is being used in the early deployments, since it takes advantage of the existing LTE infrastructure and the coverage it provides and unlocks the capacity and data rates provided by NR, while providing a seamless option to deploy 5G services with little disruption in the network. However, NSA only addresses the eMBB usage scenario, leaving the mMTC and URLLC scenarios to be addressed by the SA architecture;
- **Standalone (SA) 5G NR** - SA refers to the architecture options where only the 5G NR radio access is connected to the 5GC. A UE can directly establish a radio link with a 5G base station,

also called Next Generation NodeB (gNB), and attach to the 5GC to establish service. One of the conveniences compared to NSA is that SA is a simple solution for operators to manage and may be deployed as an independent network using normal inter-generation handover between 4G and 5G for service continuity. SA not only offers the services included in eMBB, but also in URLLC. In Figure 2.4, a more detailed schematic of the SA architecture option is presented. As depicted, the NR base stations, or gNBs, connect with each other via the Xn interface, and the RAN connects to the 5GC network using the NG interface. The core network is mainly composed by the Access and Mobility Management Function (AMF) and User Plane Function (UPF) modules, which will be described later in this section.

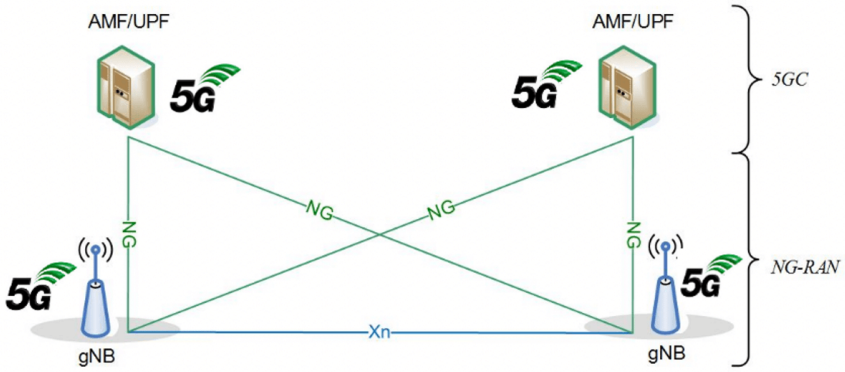


Figure 2.4: SA 5G NR Architecture [10].

2.3.2 NG-RAN Architecture

The overall architecture of the Next-Generation Radio Access Network (NG-RAN) can be seen in Figure 2.5.

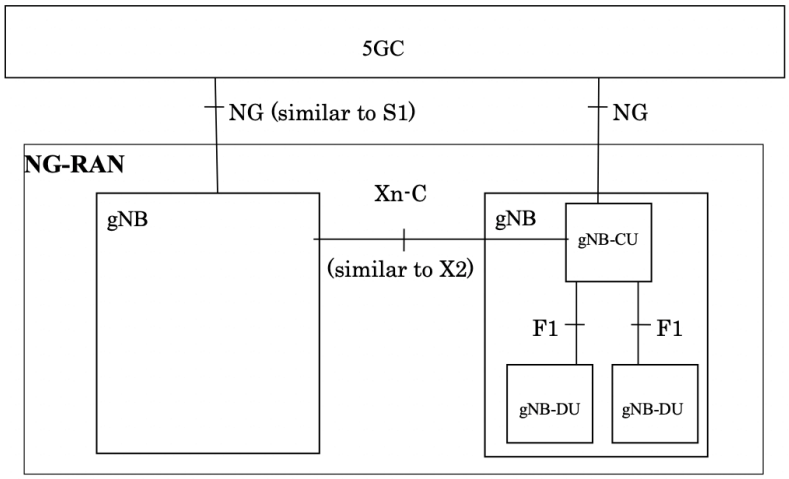


Figure 2.5: Overall NG-RAN architecture [10].

The NG-RAN consists of a set of gNBs connected to the 5GC through the NG interface. The gNB can be connected to another gNB through the Xn interface. The gNB performs tasks such as routing of

user plane data towards UPF, routing of control plane information towards AMF and support of network slicing. The gNB may be split in two basic elements:

- **Central Unit (CU)** - Handles relatively higher layers above the physical layer. The physical implementation of the CU can be via dedicated hardware or a radio-cloud implementation. The CU is connected to both the DU and to the 5G core network;
- **Distributed Unit (DU)** - Located on the cell site together with the antennas and the RF unit. Handles time-critical processing which does not tolerate the delay associated to processing away from the cell site.

One difference from the 4G RAN architecture is that, instead of having a single block, the gNB is split, which provides a flexible hardware implementation that allows scalable cost-effective solutions. Also, a split architecture allows coordination of performance features, load management, real-time performance optimization, and enables virtualized deployments.

Furthermore, with 5G, there is the possibility of using Cloud RAN (C-RAN) architectures. This enables operators to address different application requirements by having the option of physically locating storage and compute resources either at the base of the cell site, at centralized hubs hundreds of kilometers away or anywhere in between. Also, one possible solution for supporting latency-sensitive applications is by using MEC data centers that are located closer to the serving cell site. Figure 2.6 illustrates these solutions. In the classical Distributed 4G/5G RAN, the Remote Radio Unit (RU) and the Baseband Unit (BBU), which incorporates the DU and CU, are all located at the cell site. In a Distributed 5G Cloud RAN, the CU is moved to a regional data center. In the Centralized 5G Cloud RAN, the DU is put in the edge cloud, and in a Centralized 4G/5G RAN, both the DU and the CU are at the edge cloud.

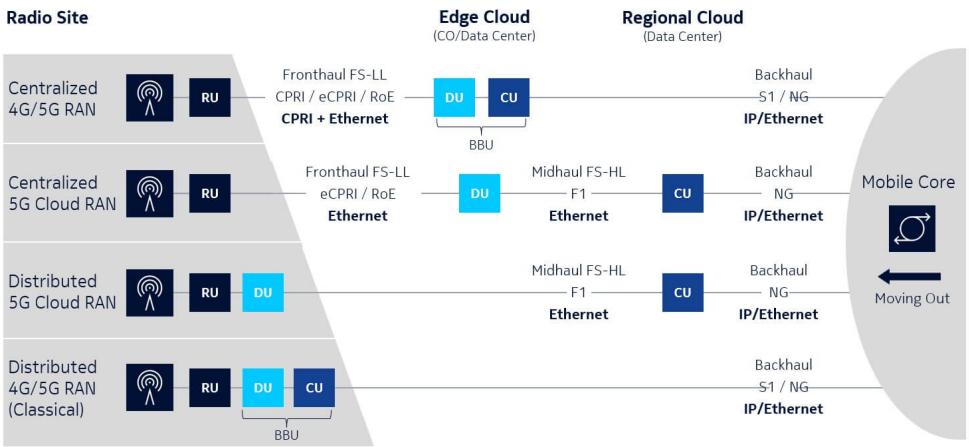


Figure 2.6: Possibilities for placing RAN functions [13].

The typical 5G RAN uses the Centralized 4G/5G RAN scenario because this is the architecture where the baseband resources are pooled so that they can be shared between base stations. This means that it is easier to adapt to non-uniform traffic and to utilize the resources more efficiently. Because fewer BBUs are required in C-RAN, the cost of network operation is decreased, due to lower energy consumption. Furthermore, new BBUs can be added and upgraded easily, improving scalability and

easing network maintenance. This architecture can also leverage the fact that BBUs from many sites are co-located in one pool in order to provide lower delays.

The fronthaul part of the network spans from the RUs to the BBU pool. The backhaul connects the BBU pool with the mobile core network. The RU, responsible for power amplification, frequency conversion, and analog to digital conversion, is co-located with the antennas. The RU is connected to the processors in the BBU pool through low latency, high bandwidth optical fiber links.

2.3.3 Network Slicing

Network slicing in 5G is intended to address services with different kinds of requirements, while at the same time improving the efficiency of the network, by enabling operators to provide networks on a case by case basis in order to meet the wide range of use cases in 5G. For example, one service may require low latency, and another may require high mobility. The slicing principles in 5G allow a UE to belong to more than a single slice, while still maintaining a single signalling connection with the network.

While legacy systems, such as 4G, host multiple services on the same architecture, network slicing enables building dedicated logical networks featuring architectures customized to the respective service, as can be seen in Figure 2.7. This provides an end-to-end system that begins at the mobile edge, continues through the mobile transport including fronthaul and backhaul segments, up to the core network.

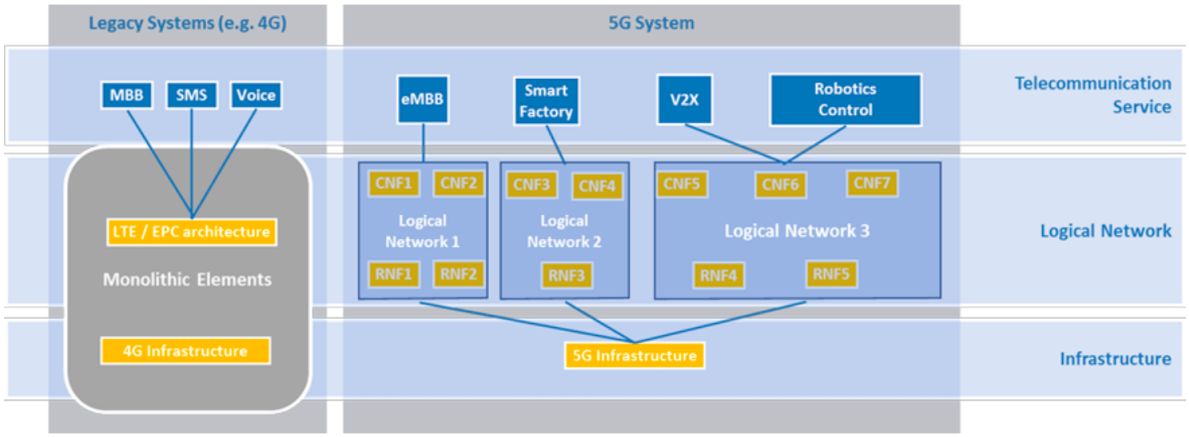


Figure 2.7: Multi-tenancy in legacy networks and slicing-enabled networks [16].

By leveraging resource abstraction technologies, the 5G architecture separates the software-based network functions from the underlying infrastructure, enabling slicing. Among these technologies is NFV and SDN, which allow different tenants to share the same general purpose hardware. These technologies enable fully decoupled end-to-end networks on top of common shared infrastructure, covering all network segments including radio networks, wire access, core, transport and edge networks.

NFV can overcome some challenges of 5G by optimizing resource provisioning of the virtual network functions for cost and energy efficiency, mobilizing and scaling virtual network functions from one hardware resource to the other, ensuring performance guarantees of virtual network functions and ensuring coexistence of virtual network functions with non-virtualized network functions.

In short, the network slice is a composition of configured network functions, network applications, and the underlying cloud infrastructure, bundled together to meet the requirements of a specific use case, such as bandwidth and latency, coupled with a business purpose. It is dedicated to a specific tenant that, in turn, uses it to provide a specific telecommunication service. Network slicing enables operators to provide programmable capabilities to Over the Top (OTT) providers and other market players without changing their physical infrastructure.

2.4 5G Physical Layer

As opposed to 4G LTE systems, latency, reliability and throughput requirements need to be jointly considered in 5G NR. In order to achieve this, changes to the physical layer architecture must be made. The physical layer in 5G has several differences from LTE. This section provides information on the 5G spectrum and radio frame, based on [5, 20–26].

2.4.1 5G Spectrum

As seen in Section 2.2, 5G radio is designed for flexible utilization of all available spectrum options. 3GPP has defined that 5G will operate in two different regions: the sub-6 GHz region, referred to as Frequency Region 1 (FR1), covering the range from 410 MHz up to 7.125 GHz, and the millimeter wave region, referred to as Frequency Region 2 (FR2), covering from 24.25 GHz up to 52.6 GHz.

While the sub-6GHz bands are heavily congested, in the millimeter wave region the spectrum can provide wide bandwidth of up to 1-2 GHz, which is necessary to achieve the desired data rates of 5-20 GHz. Millimeter wave is suited for use cases like mass events, outdoor and indoor hotspots, and fixed wireless access. Millimeter wave can also be used in cases where it is useful to offload traffic from the lower band, in order to reduce congestion.

However, millimeter wave suffers from limited coverage, which is not suitable for wide area coverage, important for use cases such as IoT and critical communication. Therefore, a combination across all bands is likely to be used: millimeter wave, mid-range frequencies, such as 2.5-5 GHz for coverage in urban areas, and frequencies below 3 GHz for wide area rural coverage, low latency and high reliability.

2.4.2 5G Waveform Format

As in 4G LTE, 5G NR has adopted a Cyclic Prefix Orthogonal Frequency Division Multiplexing (CP-OFDM) based waveform for the downlink transmission. In the uplink transmission 5G has adopted CP-OFDM, and also has an option to use Discrete Fourier Transform Spread OFDM (DFT-s-OFDM), which is useful for coverage limited scenarios.

OFDM presents several properties that make it suitable for use in 5G NR. Among these is high spectral-efficiency, which is vital to achieve extreme data rates, straightforward use with MIMO technology, robustness to synchronization errors and flexibility and scalability. As a downside, OFDM has high Peak-

to-Average Power Ratio (PAPR). A low PAPR is needed for power-efficient transmission. However, it is possible to mitigate this problem with techniques that only affect performance in a minor way.

In addition to this, OFDM supports full Frequency Division Duplex (FDD) and Time Division Duplex (TDD) modes. FDD is a duplex scheme where uplink and downlink transmissions occur simultaneously using different frequencies. TDD is a duplex scheme where uplink and downlink transmissions occur at different times sharing the same frequency. While in FDD there must exist a guard band between the sending and receiving frequencies in order to prevent receiver blocking, TDD requires a guard interval between transmission and reception times in order to allow sending signals to arrive at the destination before it starts transmitting.

2.4.3 OFDM Numerology

Unlike in 4G LTE, where only a sub-carrier spacing of 15 KHz is used, 5G NR enables flexibility of waveforms with scalable Subcarrier Spacing (SCS), according to $\Delta f = 2^{\mu} \times 15$ KHz, where μ is an integer between 0 and 4, yielding spacings between 15 KHz and 240 KHz. The choice of μ has to do with factors such as cell size, latency requirement and carrier frequency. As can be seen in Figure 2.8, wider subcarrier spacing is beneficial in situations where latency is critical, such as 360° video streaming, while narrower subcarrier spacing is suitable for outdoor coverage.

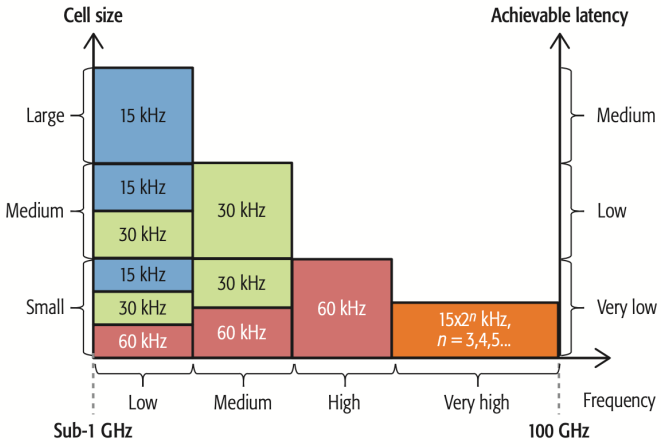


Figure 2.8: OFDM numerology for different frequencies, cell sizes and latency requirements [21].

2.4.4 5G Radio Frame Structure

In 5G NR, downlink and uplink transmissions continue to be organized into 10 ms long frames, as in 4G LTE. Each frame consists of 10 subframes of 1 ms each. The number of slots per subframe varies according to μ, as illustrated in Figure 2.9. The number of Orthogonal Frequency Division Multiplexing (OFDM) symbols per slot can change based on the subcarrier spacing and cycle prefix. We observe that an integer number of slots of one numerology fits into a slot of another numerology. This time alignment of slots is important for TDD networks to allow time-aligned uplink and downlink transmission periods.

Table 2.1 specifies the number of OFDM symbols per slot (N_{symb}^{slot}), number of slots per frame ($N_{slot}^{frame,\mu}$)

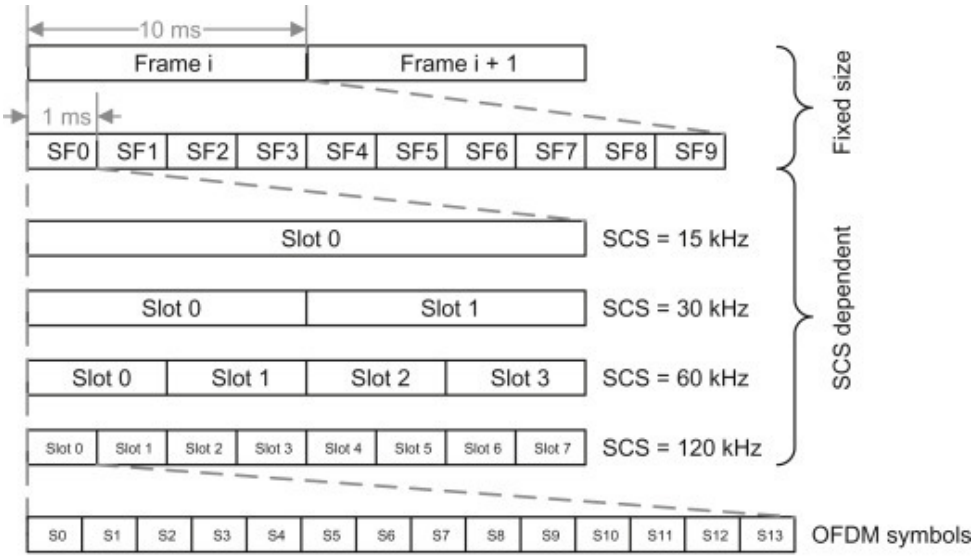


Figure 2.9: 3GPP NR frame structure [27].

and number of slots per subframe ($N_{slot}^{subframe,\mu}$), depending on subcarrier spacing and cycle prefix. The number of OFDM symbols in a subframe is $N_{symb}^{subframe,\mu}$, which is equal to $N_{symb}^{slot} \times N_{slot}^{subframe,\mu}$.

Table 2.1: Number of OFDM symbols per slot, slots per frame and slots per subframe [24].

μ	N_{symb}^{slot}	$N_{slot}^{frame,\mu}$	$N_{slot}^{subframe,\mu}$	$N_{symb}^{subframe,\mu}$	Cyclic Prefix
0	14	10	1	14	Normal
1	14	20	2	28	
2	14	40	4	56	
3	14	80	8	112	
4	14	160	16	224	
2	12	40	4	48	Extended

3GPP has also defined that mini slots with 2, 4 or 7 OFDM symbols may also be used for transmissions shorter than one slot duration. This is useful in the URLLC usage scenario, as the regular slot duration is too long or a transmission needs to be made without waiting for a new slot to start.

Table 2.2 summarizes other parameters related to the different OFDM numerologies.

Table 2.2: Scalable OFDM numerology parameters [21].

Subcarrier spacing	**15 KHz**	**30 KHz**	**60 KHz**	**120 KHz**
OFDM symbol duration	66.67 μs	33.33 μs	16.67 μs	8.33 μs
Cyclic prefix duration	4.69 μs	2.34 μs	1.17 μs	0.59 μs
OFDM symbol including CP	71.35 μs	35.68 μs	17.84 μs	8.92 μs
Number of OFDM symbols per slot	14	14	14	14
Slot duration	1000 μs	500 μs	250 μs	125 μs

2.4.5 5G Resource Grid

A resource grid is formed by one subframe in the time domain and a carrier bandwidth in the frequency domain, as can be seen in Figure 2.10. There is one resource grid for each transmission direction (downlink or uplink), numerology configuration and antenna port. A Resource Element (RE) is its smallest unit, and is constituted by the combination of one subcarrier in the frequency domain and one OFDM symbol in the time domain.

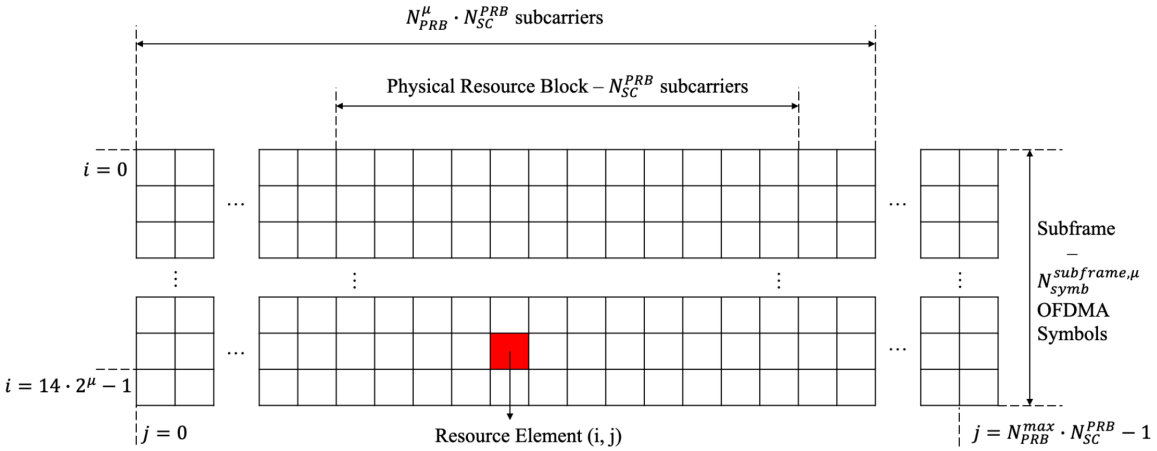


Figure 2.10: Resource grid in subframe level [24].

REs are grouped into Physical Resource Blocks (PRBs). Each PRB consists of N_{SC}^{PRB} subcarriers (fixed to 12). The number of PRBs varies depending on the used numerology and bandwidth. The number of PRBs for a given configuration is presented in Table 2.3.

Table 2.3: Number of physical resource blocks for a given configuration for FR1 [20].

SCS [kHz]	Bandwidth [MHz]											
	5	10	15	20	25	30	40	50	60	80	90	100
15	25	52	79	106	133	160	216	270	N/A	N/A	N/A	N/A
30	11	24	38	51	65	78	106	133	162	217	245	273
60	N/A	11	18	24	31	38	51	65	79	107	121	135

2.5 5G Performance

This section analyzes 5G from a performance standpoint. There are several performance metrics that 5G is targeting, such as mobility, energy efficiency, reliability and data volume. This section focuses on the capabilities of 5G in terms of data rates, latency and transmission efficiency, based on [5, 6, 28–32].

2.5.1 5G Peak Data Rates

5G is designed to deliver peak data rates up to 20 Gbps - up from the 1 Gbps peak data rate achieved with LTE Advanced. Naturally, this data rate can only be achieved with the most ambitious implementa-

tions. In practice, the data rate depends on a variety of factors, such as bandwidth, antenna configuration, modulation order, and others.

In 5G NR, the approximate data rate can be calculated in Mbps by

$$\text{rate} = 10^{-6} \cdot \sum_{j=1}^{J} \left(v_{Layers}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{max} \cdot \frac{N_{PRB}^{BW(j),\mu} \cdot 12}{T_s^{\mu}} \cdot \left(1 - OH^{(j)}\right) \right), \tag{2.1}$$

where:

- J is the number of aggregated component carriers in a band or band combination. Carrier Aggregation (CA) is a technique introduced by LTE Advanced that allows the usage of more than one Component Carrier (CC) in order to expand the available bandwidth;
- R_{max} is the maximum coding rate. Coding rate is defined as the ratio between the number of useful bits and total transmitted bits and is defined by the used Modulation Coding Scheme (MCS). This topic is addressed with more detail in subsection 2.5.2;
- For the j^{th} CC:
  - $v_{Layers}^{(j)}$ is the maximum number of layers;
  - $Q_m^{(j)}$ is the maximum modulation order;
  - $f^{(j)}$ is the scaling factor, which can take the values 1, 0.8, 0.75 and 0.4;
  - μ is the numerology as defined in Subsection 2.4.3;
  - T_s^{μ} is the average OFDM symbol duration for numerology μ, i.e., $T_s^{\mu} = \frac{10^{-3}}{14 \cdot 2^{\mu}}$. Normal CP is assumed;
  - $N_{PRB}^{BW(j),\mu}$ is the maximum PRB allocation in bandwidth $BW^{(j)}$ with numerology μ, as defined in Subsection 2.4.5;
  - $OH^{(j)}$ is the overhead and takes the value 0.14 for FR1 for downlink and 0.18 for FR2 for downlink.

As an example, a configuration of 100 MHz bandwidth, 30 kHz SCS, 4x4 MIMO and 256-QAM modulation has a theoretical peak data rate of 2.3 Gbps. The practical data rate is lower for various reasons, such as guard bands, control channel overhead, signal levels, among others.

2.5.2 5G Modulation Coding Scheme

The Modulation Coding Scheme (MCS) defines the proportion of useful bits in relation to total transmitted bits which can be carried by one Resource Element (RE). The Adaptive Modulation and Coding (AMC) technology allows the appropriate MCS to be dynamically selected as a function of radio signal quality. A lower MCS reduces the transmission rate, but it also reduces the Block Error Rate (BLER) in a poor channel quality environment, ensuring the accuracy of the data received by the UE.

The MCS defines two aspects: the type of modulation used and the code rate. The modulation defines how many bits can be encoded into an OFDM symbol. In 5G NR, Quadrature Phase-Shift Keying

(QPSK), 16-Quadrature Amplitude Modulation (QAM), 64-QAM and 256-QAM modulation schemes are supported. The corresponding modulation modulation orders are 2, 4, 6 and 8, respectively.

The code rate is the ratio between useful bits and total transmitted bits (useful and redundant bits). The function of the redundant bits is to enable Forward Error Correction (FEC). As such, when the channel quality is poorer, a lower coding rate is used, meaning more redundant bits are transmitted.

In order to select the appropriate MCS, the radio channel quality has to be estimated. This is achieved with the Channel Quality Indicator (CQI). The CQI is a metric, periodically transmitted from the UE to the Base Station (BS), which indicates the estimated highest modulation and code rate at which the BLER does not exceed 10%. This is achieved taking into account the Signal-to-noise Ratio (SNR) of the channel, among other metrics. The higher the value of the CQI, the higher the achievable efficiency of the transmission, allowing for more useful bits to be transmitted with each PRB.

The achievable MCS for each CQI is shown in Tables 2.4 and 2.5. The 64-QAM table is used when the BS or the UE do not support 256-QAM or in poor radio conditions where 256-QAM decoding is unsuitable. The 64-QAM table is also suitable for applications where reliable data transfer is a requirement.

Table 2.4: Efficiencies for normal channel [29].

CQI Index	Modulation	Code rate (×1024)	Spectral efficiency
0	(out of range)		
1	QPSK	78	0.1523
2		193	0.3770
3		449	0.8770
4	16-QAM	378	1.4766
5		490	1.9141
6		616	2.4063
7	64-QAM	466	2.7305
8		567	3.3223
9		666	3.9023
10		772	4.5234
11		873	5.1152
12	256-QAM	711	5.5547
13		797	6.2266
14		885	6.9141
15		948	7.4063

Table 2.5: Efficiencies for poor channel [29].

CQI Index	Modulation	Code rate (×1024)	Spectral efficiency
0	(out of range)		
1	QPSK	78	0.1523
2		120	0.2344
3		193	0.3770
4		308	0.6016
5		449	0.8770
6		602	1.1758
7	16-QAM	378	1.4766
8		490	1.9141
9		616	2.4063
10	64-QAM	466	2.7305
11		567	3.3223
12		666	3.9023
13		772	4.5234
14		873	5.1152
15		948	5.5547

2.5.3 5G User Plane Latency

In order for 5G to support interactive application such as 360° video streaming, low latency communications is essential. Table 2.6 breaks down the best-case scenario latency components in LTE and 5G. It is important to note that these values are dependent on using shorter Time Transmission Intervals (TTIs) and mini-slot transmission. It is also assumed that 5G can use grant-free transmission without

any scheduling delay. For the transport and core, it is assumed the content is close to the base station.

Table 2.6: Best-case scenario latency components in LTE and 5G [5].

Delay component	LTE Release 8	5G with short TTI transmission
Downlink transmission [ms]	1	0.125
Uplink transmission [ms]	1	0.125
Downlink frame alignment [ms]	0.5	0.07
Uplink frame alignment [ms]	0.5	0.0625
Scheduling [ms]	3-18	-
UE processing [ms]	2.3	0.2
BTS processing [ms]	2	0.2
Transport and core [ms]	1	0.1 (for local content)
Total [ms]	**11-28**	**0.9**

2.6 Multi-Access Edge Computing in 5G

This section covers a technology widely known as mobile edge computing and is based on the literature in [9, 33–35]. However, it should be noted that in order to further complement the capabilities of mobile edge computing with heterogeneous access technologies, such as 4G, 5G, Wi-Fi and fixed connection, the European Telecommunications Standards Institute (ETSI) changed the name of mobile edge computing to Multi-Acess Edge Computing (MEC). This change was made to reflect the fact that MEC servers can be deployed at different locations within the architecture.

MEC is a key technology in 5G. It is defined by ETSI as a technology that provides an IT service environment and cloud computing capabilities at the edge of the mobile network, within the RAN and in close proximity to mobile subscribers. More specifically, it can host compute-intensive applications, process large data before sending to the cloud and offer context-aware services. As such, MEC enables a wide variety of applications where a real-time response is strictly required, such as driverless vehicles, VR/AR, robotics and immerse media.

MEC use cases can be divided into consumer-oriented services, operator and third-party services and network performance and QoE improvements. The consumer-oriented services use case aims to bring direct benefits to users through running computation-heavy and latency-sensitive applications at the network edge. This is where the AR/VR applications are included.

These emerging applications are anticipated to be some of the most demanding so far in wireless networks. MEC would benefit these applications with longer battery lifetime, a powerful set of computing and storage resources, and low end-to-end latency.

2.6.1 MEC Fundamentals

Among the factors that have been driving the demand for MEC is low-latency computing, as this is a fundamental metric for network performance and is required by many applications, such as VR. MEC is

mainly characterized by:

- **On-premises** - The edge is local, meaning that it can run independently from the rest of the network, while having access to local resources;
- **Proximity** - Edge computing is close to the UEs, being useful to capture key information for analytics and big data;
- **Lower latency** - As edge services run close to the UEs, this considerably reduces latency, making them a key enabler to latency-critical applications;
- **Location awareness** - A local service can leverage low-level signalling information to determine the location of each connected UE;
- **Network context information** - Real-time network data can be used to offer context-related services in order to improve the Quality of Service (QoS).

However, MEC poses several challenges. Some of these challenges are:

- **Distributed resource management** - Due to finite resources, the channel can become congested and there can be scarcity of computing capacity. Although the centralized approach seen in Subsection 2.3.2 offers benefits in performance, its high computational complexity can make it unsuitable for distributed MEC systems;
- **Reliability and mobility** - User mobility can lead to frequent handovers between base stations, which affect the network performance. Users with high mobility may not be able to receive the result since they can move out of the coverage of the MEC servers;
- **Network integration and application portability** - Since MEC servers can be deployed in several points within the RAN, their seamless integration into the underlying architecture is a challenge;
- **Coexistence of distributed MEC and centralized cloud** - Cloud data centers and MEC servers each have their qualities and shortcomings. Cloud data centers have vast computing resources, while MEC servers can greatly reduce the end-to-end delay. It is important to implement MEC distributing the computations between the two.

Chapter 3

360° Video Streaming

This chapter provides a literature review of 360° video streaming. The topic is contextualized in Section 3.1. Section 3.2 analyzes the challenges associated with this technique. A typical 360° video streaming framework is provided in Section 3.3, while Section 3.4 goes further into its transmission aspect. Section 3.5 provides the available approaches to this method. Section 3.6 explains why MEC is a pivotal enabler for 360° video services. Section 3.7 introduces the topic of user QoE. An overview of the state-of-the-art in MEC-assisted 360° video streaming is given in Section 3.8.

3.1 Introduction

Recent years have seen a rapid increase in the pervasiveness of VR and AR applications. Cisco [3] identifies the growth of mobile data traffic generated by these applications as one of the top global mobile networking trends. Due to the rollout of 5G and the advancements in technologies such as MEC, this type of traffic will see a 63% Compound Annual Growth Rate (CAGR) between 2017 and 2022, from 22 petabytes per month in 2017 to 254 petabytes per month in 2022 [3].

One of the main applications of VR is 360° video which, supported by the growing number of existing VR headsets and availability of omnidirectional cameras, has created significant interest in this type of content. 360° video presents an improvement over the traditional 2D video in the sense that it provides an immersive environment that 2D video cannot provide, allowing the user to freely choose the viewing direction.

360° videos are spherical videos in which the user can choose the section of the sphere that is in their Field of View (FoV). A 360° video is typically presented to the user by means of a HMD. The HMD is equipped with sensors that detect which direction the user is facing and it uses this information to deliver the corresponding region of the 360° video, known as the viewport [36]. 360° videos may also be presented to the user by means of a regular screen, with the user choosing the viewport orientation using a mouse or keyboard.

In addition to the growing availability of devices to produce and consume 360° video, video sharing websites and social media platforms have started allowing the users to share this type of content. Therefore, creating, distributing and consuming 360° videos is a straightforward process, which explains why the streaming of such videos is one of the main VR applications [4].

3.2 360° Video Streaming Challenges

In comparison to traditional 2D video, 360° video represents a new paradigm. This mainly has to do with the fact that, while 2D video content is the same for every user, 360° video gives the possibility of choosing the viewing direction to the user. This means that the delivery of high quality 360° video over mobile networks poses significant challenges when compared to 2D video due to the strict bandwidth and latency requirements. This section covers such challenges, based on the literature in [4]:

- **High bandwidth requirements** - The required bandwidth to stream a 360° video is an order of magnitude greater when compared to a conventional 2D video. While a 2D video stream with a resolution of 4K requires about 25 Mbps of data rate, a high quality 360° video can require up to 200 Mbps. This aspect is made more challenging when a multiuser scenario is being considered;
- **Large storage requirements** - The increase in data rate also means that 360° video takes up more storage space. Furthermore, the content needs to be stored in multiple formats in order to be compatible with multiple devices at different quality levels;
- **Low motion-to-photon latency** - In order to avoid motion sickness, when the user turns their head, the delay in rendering the new view should be in the order of 20 milliseconds. Although 5G networks are expected to be able to reach a latency of 1 ms, the end-to-end latency also includes application processing times and transport delays;
- **Complex view adaptation** - 360° video presents the challenge of adapting the video to the viewport as the user moves their head. One solution is to deliver the full 360° panorama to the user, with the area covered by the viewport being rendered on-device. However, this approach wastes bandwidth since there is a portion of the video that is not covered by the viewport. Other solution is to split the video into several regions, called tiles, with each being available in different qualities to support Adaptive Bitrate (ABR) streaming. The challenge in this solution is to predict the viewing direction and always have the correct tiles buffered, ready to be played without stalls;
- **Video QoE** - Unlike the case of traditional 2D video, the factors that influence 360° video QoE are not completely understood, and may require the development of new metrics that reflect the perceived quality with accuracy.

These challenges make the current video delivery architectures unsuitable for the 360° video case, unless quality compromises are made. Critically, the transmission bandwidth needs to be reduced and the network infrastructure needs to be optimized in order to reduce the end-to-end latency and manage the available resources in multiuser scenarios.

3.3 360° Video Streaming Framework

This section describes the main processes of the typical transmission chain, from the moment a 360° is acquired to the moment it is seen by a user, based on the literature from [4, 36–42]. Figure 3.1 illustrates these processes. Each process is described below.

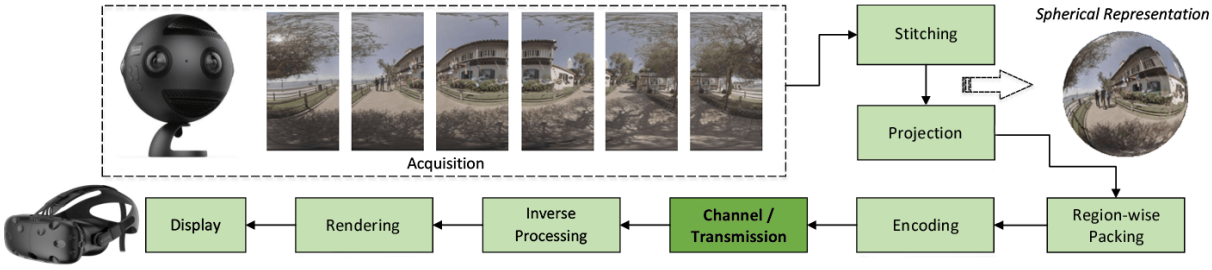


Figure 3.1: Architecture of the 360° video transmission chain (adapted from [43]).

Acquisition

A centric camera system is used to record the 360° scene of an environment. This capture is usually done with a rig of multiple centrally mounted cameras, which are calibrated and time synchronized. Each camera points in a different direction, in order to acquire a 2D image corresponding to a portion of the spherical view around it. There must be enough coverage of the view to capture the entire panorama.

Stitching

The captured images from different directions need to be manipulated in order to obtain a single seamless representation of the omnidirectional scene, as seen in Figure 3.1. This process is called stitching. It is composed of several steps, such as calibration, registration and compositing. The generated omnidirectional image is a non-planar frame, with a spherical representation usually used.

Projection

The spherical frame needs to be projected onto a planar frame in order to be effectively coded and transmitted. Analogously to projecting the globe onto a map, this is a process that causes distortion on the image. Different projection schemes cause different kinds of distortion. In 360° video applications, Equirectangular Projection (ERP) and Cubemap Projection (CMP) are two popular projection schemes, which are represented in Figure 3.2.

(a) Equirectangular projection (b) Cubemap projection

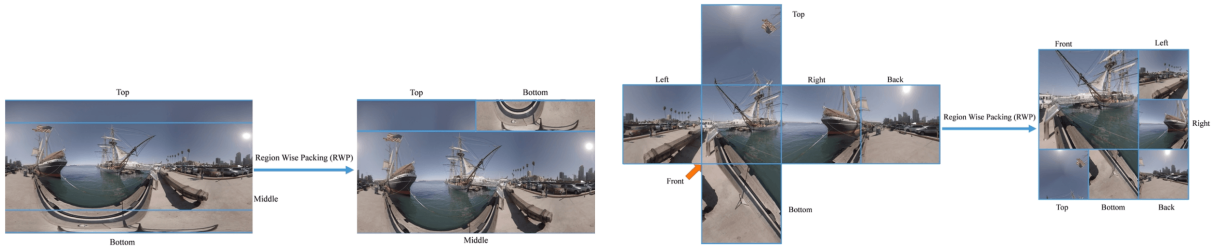
Figure 3.2: Projection schemes of 360° video [36].

The ERP flattens the sphere onto a rectangular plane. This projection scheme produces more distortion the farther a region is from the equator of the sphere, with the poles having significant distortion due to being stretched to fill the top and bottom of the projection. This reduces the encoding efficiency, as it is more expensive to encode these regions compared to the others. In the CMP, the pixels of

the sphere are mapped onto a six-sided cube. Each side contains a 90° FoV. This projection scheme creates less distortion in comparison to the equirectangular projection. However, the planar frame may present artifact along the edges of the square, due to the discontinuities.

Region-wise Packing

The region-wise packing process may be used prior to encoding in order to resize, reposition, rotate and mirror any region of the projected planar frame. Through this process, it is possible to distinguish regions in the projected frame with different quality levels, providing additional compression by increasing the sample density of some regions, as illustrated in Figure 3.3. This process can be used with viewport-independent projection schemes to downsample the less important regions, such as the top and bottom in equirectangular projections or all faces of the cubemap projection except for the front, or with viewport-dependent projection schemes to generate viewport-specific video.



(a) Equirectangular projection (b) Cubemap projection

Figure 3.3: Region-wise packing for different projection schemes [36].

Encoding

Since the projection process generates planar frames, a 2D video codec can be used to encode them. The current generation of 4K 360° videos require 10-50 Mbps in terms of bitrate, while next-generation 360° videos may require up to 200 Mbps. Therefore, efficient compression is essential.

Modern video codecs work by exploiting spatial and temporal redundancies in the video in order to achieve the desired compression. A coded video bitstream is composed by a sets of frames known as Group of Pictures (GOP). Each GOP contains a fixed number of frames (15 is a typical value) which can be of one of three types: I-Frame, P-Frame and B-Frame. A frame is spatially divided into macroblocks. A GOP includes an I-Frame at the start and all the subsequent frames until the next I-Frame. The different frame types are defined as:

- **I-Frame** - An Intra-coded frame is coded using only information from itself, i.e. with no reference to other frames. As such, an I-Frame is only consisted of macroblocks that use spatial prediction. Although they have a lower compression factor, I-frames can be independently encoded and decoded, meaning they can help recover from bitstream errors, which justifies their usage;
- **P-Frame** - A Predictive-coded frame is coded using motion compensated predictions from a previous I or P-Frame. This means that a P-Frame allows macroblocks to be compressed using both spatial prediction and temporal prediction in relation to frames from the past;

- **B-Frame** - A Bidirectionally-predictive-coded frame is coded using motion compensated predictions from a past I or P-frame or future I or P-frame. It is similar to a P-Frame except that it can use temporal prediction in relation to both future and past frames. Due to this capability, B-frames are the most efficient in reducing the size of the frame, at the expense of computational complexity.

Figure 3.4 depicts the dependence the various frame types have on each other.

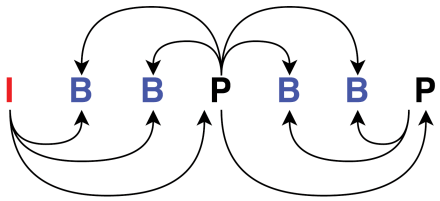


Figure 3.4: I/P/B-frame referencing (adapted from [41]).

The decoded quality of the video is dependent by the Quantization Parameter (QP) used in the encoding, which regulates the quantity of spatial detail saved. When a high QP is used, the resulting bitrate is decreased, as the associated quality will be lower.

Currently, the used codecs for 360° video are Advanced Video Coding (AVC)/H.264 or High-Efficiency Video Coding (HEVC)/H.265. HEVC/H.265, which is a more recent codec, reduces the video bitrate by 50% compared to AVC/H.264, while maintaining the used QP and therefore the perceived quality.

Channel/Transmission

The bitstream generated by the encoder is either stored in a server or sent directly to the client, depending on whether it is a live streaming or an on-demand application. The bitstream may be sent via a fixed or wireless communication channel. One popular standard for delivering video content using OTT services is MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH), due to its simplicity and compatibility with network infrastructure. This protocol is further described in Section 3.4.

Inverse Processing

The 360° video content is decoded, obtaining the planar frame. For the rendering process, a spherical representation is typically used. As such, the planar frame is projected into a sphere, by applying the corresponding inverse transformation from the projection process.

Rendering

The images that are presented to the user are a section of the spherical representation that is in their FoV. This section is projected onto a 2D plane, known as the viewport. Out of the several projection schemes that can be used, the perspective projection is the most popular. Nowadays, client-based rendering is almost always used. However, this presents a wastage of resources due to the need to process parts of the content that are outside of the viewport. Another option is cloud-based rendering. With this solution, only the requested viewports are streamed to the users.

Display

The output of the rendering process is a 2D image that can be presented on a display. This display can be a standard one, with the viewing direction being controlled by a mouse, or it can be a HMD, which detects which direction the user is facing in order to compute the corresponding viewport.

Using a regular display does not offer an immersion sensation to the user. However, almost every user already owns one of these devices, be it a smartphone or a computer. This provides an accessible entry point to the 360° video streaming experience. The HMD aims at providing the best experience to the user, tracking their head orientation and displaying the corresponding images. These devices have two displays, one for each eye, which allows for stereoscopic omnidirectional video.

3.4 MPEG-DASH

As stated in Section 3.3, the bitstream generated by the encoder needs to be sent to the client over the communication channel. Due to fluctuations in the channel's quality and availability, delays and loss of data may happen, impairing the viewing experience. In order to solve this problem, ABR techniques are used in order to adapt the transmitted video quality (i.e. bitrate) to the user's available bandwidth. One popular standard to incorporate this technique is the MPEG-DASH protocol. This section provides an overview on the MPEG-DASH protocol, based on [4, 38, 44, 45].

3.4.1 MPEG-DASH Framework

The MPEG-DASH stream is divided into temporal segments, encoded with multiple bitrates or spatial resolutions. The MPEG-DASH framework is illustrated in Figure 3.5. In this example, the video is encoded in three qualities. The client requires the different qualities according to their available bandwidth.

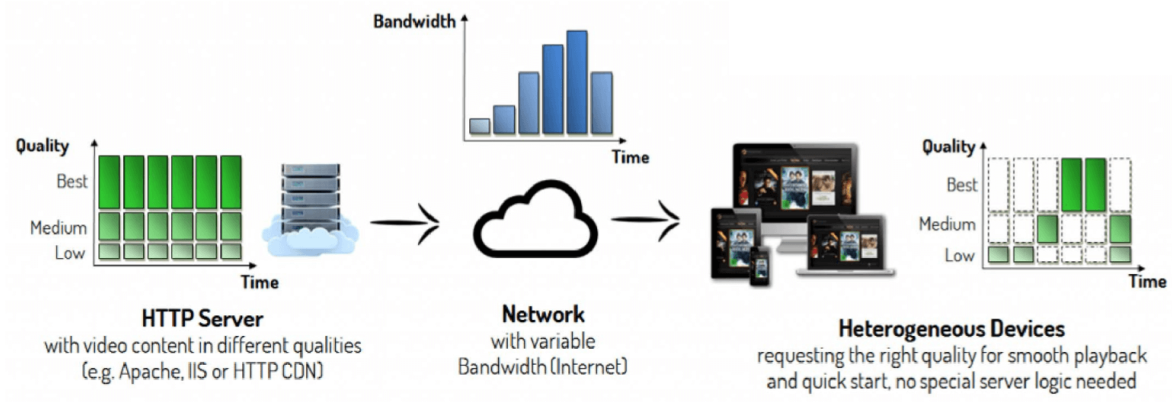


Figure 3.5: MPEG-DASH video framework [4].

The MPEG-DASH standard serves as the foundation for 360° video streaming, by leveraging the DASH video framework for bitrate adaptation. Since it is built upon the Hypertext Transfer Protocol (HTTP) protocol, its integration within applications is straightforward. HTTP also has the advantage of being a stateless protocol. The connection between the user and the server ends when a request is answered.

This allows for a better usage of server resources when many users are connected.

3.4.2 MPEG-DASH Scope

Figure 3.6 presents the scope of the MPEG-DASH standard. The multimedia content is captured and stored on an HTTP server and the delivery occurs over HTTP. Stored in the server is the Media Presentation Description (MPD), which is a manifest of the available segments, which are the actual videos bitstreams in the form of chunks. The client first requests the MPD, through which it learns about the video metadata, such as the available media types, resolutions, bandwidths, among others. Based on this, the client then selects the appropriate content and starts streaming it by making HTTP requests. The client is in control to select which bitrates and resolutions it requests and to switch between qualities depending on its available resources.

The MPD's top level consists of one or more temporal periods, where each period has a starting time and duration and consists of one or more adaptation sets. The adaptation sets provide information on the available media streams, which usually have multiple representations. A representation is an encoded segment of the same content, varying by bitrate, resolution, or others. The client may dynamically switch between representations within an adaptation set in order to adapt to network conditions. Each representation contains one or more segments, which are the media stream chunks in temporal sequence. Each segment is characterized by a server address that enables the client to request it via HTTP.

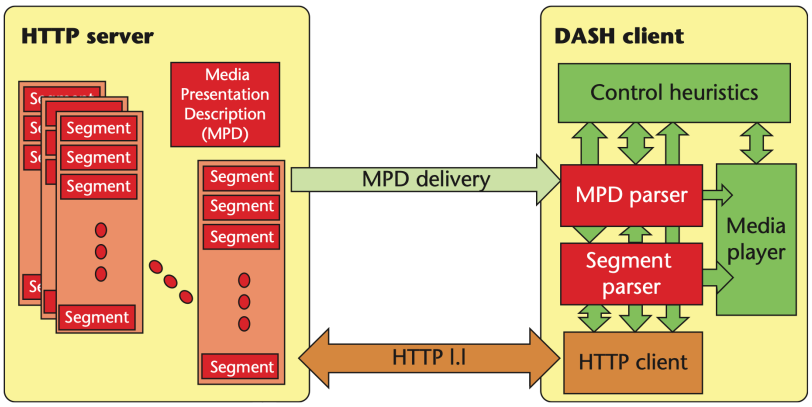


Figure 3.6: Scope of the MPEG-DASH standard [44].

3.5 360° Video Streaming Schemes

This section provides information on the various 360° video delivery schemes, based on the literature from [38, 39, 46–50].

In addition to the features offered by MPEG-DASH, presented in Section 3.4, 360° video streaming poses specific challenges that MPEG-DASH alone does not address. For example, it would be highly useful for a client in a 360° video streaming scenario to be able to request different spatial parts of the video with different representations, in order to take advantage of the fact that the viewport only covers a fraction of

the viewing sphere. This way, it can save bandwidth by requesting lower quality representations in areas of the video where high quality is not needed, and have more flexibility to adapt to the varying channel conditions. This approach to streaming is known as Viewport Adaptive Streaming (VAS).

MPEG-DASH is a flexible standard that allows the addition of new features. The Spatial Representation Description (SRD) feature extends the MPD of MPEG-DASH by describing spatial relationships between associated pieces of video content. With SRD, the client can request the different parts of the video at a resolution that is relevant to the user experience.

With the aim of standardizing the storage and delivery format for 360° video content and avoid market fragmentation, the Moving Picture Experts Group (MPEG) has also developed the Omnidirectional MediA Format (OMAF), a standard specifically for VR. It builds upon and is compatible with already existing standards, such as codecs (HEVC), file formats (ISOBMFF) and delivery signalling (DASH). It also includes metadata information for encoding, projection, region wise packing and viewport orientation.

OMAF includes three different streaming schemes: viewport-independent streaming, viewport-dependent streaming and tile-based streaming. These schemes are presented in the following subsections.

3.5.1 Viewport-independent Streaming

This scheme, also known as monolithic streaming, streams the entire scene in uniform quality using a viewport-agnostic projection scheme, such as ERP or CMP, described in Section 3.3. Using this scheme, no modifications to the MPEG-DASH framework are needed. The bitrate adaptation of the DASH client is made in similar manner to a 2D video, meaning that the representation requests for upcoming segments are made only according to the available bitrate. While the deployment for this approach is simple, a large part of the bandwidth is wasted on content that is not shown to the user.

3.5.2 Viewport-dependent Streaming

Viewport-dependent streaming is a type of VAS and it aims at reducing the bandwidth usage by performing adaptation, based not only on the available bitrate but also on the user's viewing orientation. As such, the server needs to make different adaptation sets available, each emphasizing the area of the video associated with the user's viewing orientation. Each adaptation set encodes the area of the viewport with more bits, meaning it will have a higher quality than the rest of the frame.

This can be achieved using viewport-dependent projections, such as the Offset Cubemap (OCM) projection, or by applying region-wise packing to a viewport-independent projection, with a higher resolution for regions representing the requested viewport.

With this approach, it is possible to use less bandwidth compared to viewport-independent streaming. However, the server needs to store several versions of the same content, each representing a different viewing direction. As such, in addition to being less cache-efficient, more server storage is required.

Figure 3.7 represents a viewport-dependent streaming solution, based on Quality Emphasis Regions

(QERs). A QER is a region in the video which has higher quality than the other regions. In this example, the server offers 3 different adaptation sets (one for each QER), each containing 2 representations at different bitrates. The image depicts the requested representations along a streaming session where the available bandwidth and the user's head orientation vary over time.

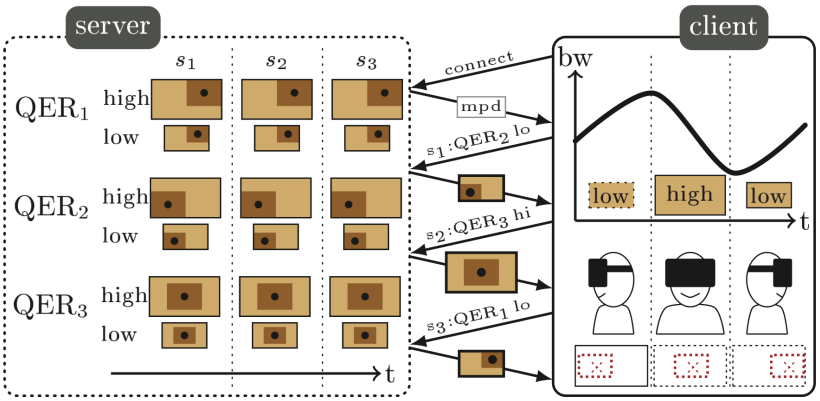


Figure 3.7: Viewport-dependent video delivery system [46].

3.5.3 Tiles-based Streaming

In addition to being temporally segmented, 360° video can further be spatially partitioned into rectangular tiles in order to independently adjust the quality of each. With this approach, the number of versions available at the server can be lower than with viewport-dependent streaming because they only depend on the granularity of the tiles and the number of different qualities. Therefore, the required storage capacity is lower. As mentioned in the introduction of this section, MPEG-DASH already supports the signalling of spatially segmented content via the SRD feature.

The client requests the tiles corresponding to the viewing direction with a high quality, and the others with a low quality. Comparing to viewport-dependent streaming, this approach also has the benefit of not depending on the FoV of the user device. Larger FoV devices can be supported simply by retrieving more high quality tiles. Figure 3.8 illustrates different possible tiling strategies for two consecutive segments. The red rectangle represents the viewport. With a full delivery strategy, all tiles are requested, the ones covered by the viewport with a higher quality than the others. In partial delivery, the tiles not covered by the viewport are not requested at all, enabling higher bandwidth savings.

One issue to consider with tiles-based streaming is the tile layout. While larger tiles provide improved coding efficiency, they offer less flexibility for viewport selection. Smaller tiles provide more flexibility in terms of selecting the ideal set for a given viewport at the expense of lower coding efficiency because the temporal and spatial correlations cannot be fully exploited, as they are coded independently.

OMAF recommends the tiles layout represented in Figure 3.9(a), using ERP. The differentiation between high and low quality tiles is made by encoding low quality tiles with half the spatial resolution of high quality tiles. Therefore, the server needs to store two versions of each tile for each representation level. Figure 3.9(b) shows the number of high quality tiles needed to cover the entirety of a viewport with a

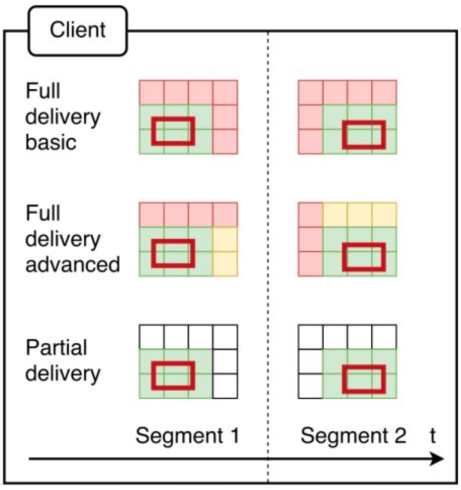
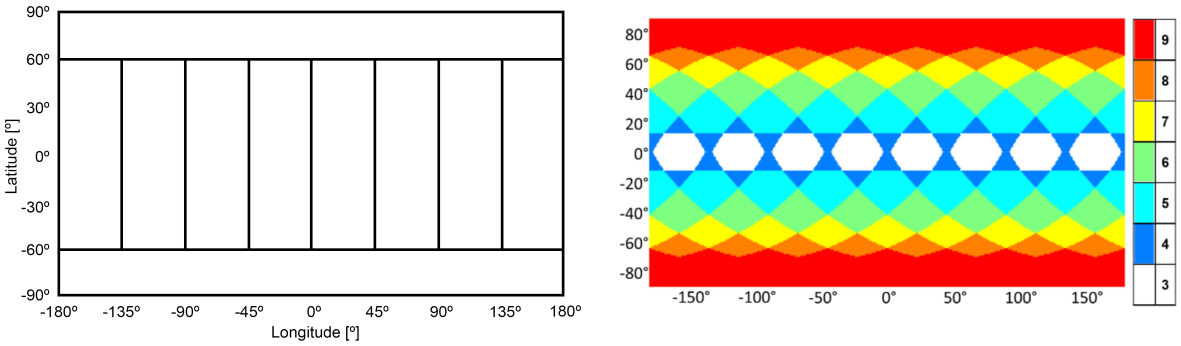


Figure 3.8: Different tiling options for adaptive 360° video streaming [36].

96° vertical and horizontal FoV, with the viewport centered at each set of coordinates.



(a) Tile layout recommendation in OMAF Annex D.6.3 [51].

(b) Number of tiles covered by the viewport at each viewport center [52].

Figure 3.9: OMAF tiles layout.

Although a tiles-based streaming scheme is advantageous from the bandwidth savings and lower server storage requirements standpoint, they have strong low latency requirements, especially in the case of partial delivery, in order to deliver the requested segments in time for the user to watch them. Otherwise, the user's head motion could lead to blank areas being rendered because of the corresponding tiles not being yet delivered.

3.5.4 Viewport-Only Streaming

Although not included in OMAF, a viewport-only streaming scheme is a solution where the client sends a request to the server for the exact viewport to display to the user. Using this scheme, only the bandwidth required to transmit the pixels inside of the viewport is necessary. However, this comes at the expense of the strictest latency requirement and the highest complexity at the server of all the presented schemes, since it needs to process a personalized viewport for each connected user. Also, some modifications and adjustments need to be made considering the typical streaming framework presented in Section 3.3.

First, it becomes clear that this streaming scheme requires the inverse processing and rendering steps to be performed at the server instead of the client, because the client needs to receive a viewport that is ready to display to the user. It is also necessary to include an extra step at the server in order to re-encode the rendered viewport to allow for the transmission.

Furthermore, this scheme also requires the transmission scheme to be changed, as using the regular implementation of MPEG-DASH becomes impossible. This is due to the fact that MPEG-DASH delivers segments that are typically between 1 and 5 seconds in length. However, since this scheme requires each frame to be personalized for each user, and knowing their exact viewport trajectory for the length of each segment a priori is impossible, it becomes unfeasible to pre-process the video sequence and have segments with these time periods ready to deliver. As such, this processing needs to made in real time, because otherwise the disparity between the pre-processed viewport and the actually requested viewport would be too large.

Assuming a video with 25 frames per second, each frame has a period of 40 ms. This is the maximum time period available for the client to read the motion sensor, send a viewport request to the server, which needs to travel through the network, and for the server to process this request, compute the viewport, encode it, and send it back. The time required to decode and refresh the display with the new viewport is also relevant. In order for this to be achieved, a protocol designed for real time communications may be used, such as the Real-time Messaging Protocol (RTMP).

In order to optimize the latency, another modification can be made to the encoding process described in Section 3.3. Typically, the profiles used in the AVC and HEVC codecs, such as the Random Access Profile (RAP), include the usage of B frames in order to increase the compression efficiency as much as possible. However, B frames are dependent on both past and future I/P frames in order to be decoded, which means that there must be a small buffer to reorder the frames for correct playback. Assuming, as illustrated in Figure 3.4, that there are two B frames between every I/P frame, the buffer would need to keep at least three frames, which would add 120 ms of latency, surpassing the maximum latency requirement. As such, only I/P frames can be used, for example with the Low Delay P Profile (LDPP), even though this comes at the expense of slightly higher video bitrates.

3.5.5 Schemes Comparison

As illustrated throughout this section, each streaming scheme presents a clear trade-off between the required minimum bandwidth to stream the 360° video content and the required maximum latency for such stream to be transmitted in time for the video to be correctly displayed. Table 3.1 presents a qualitative assessment of these requirements for each of the presented streaming schemes.

From Table 3.1 it becomes clear that the bandwidth savings made possible by using progressively less bitrate-intensive schemes come at the expense of requiring very low end-to-end latency times, in order for it to be possible for the server to respond to the requests made by the clients in time for the segments to be displayed when the user turns their head.

Table 3.1: Qualitative assessment of different streaming schemes requirements.

Streaming scheme	Bandwidth requirements	Latency requirements
Viewport-independent	High	Low
Viewport-dependent	Medium	Medium
Tiles-based full delivery	Medium	Medium
Tiles-based partial delivery	Medium/Low	Medium/High
Viewport-only	Low	High

This is not always feasible when the clients are served from geographically far away locations, especially when the number of concurrent users starts to increase. A further overview of the delivery architecture and the ways in which new solutions offered by 5G, such as MEC, can improve this issue is given in Section 3.6.

3.6 360° Video Streaming Delivery

The delivery of high-quality 360° video is expected to be leveraged by 5G networks. The traditional network architecture, with centralized cloud computing and storage, is not suitable for the delivery of this type of content because it does not meet its low latency and high bandwidth requirements. As such, MEC is regarded as a pivotal enabler for 360° video services. For example, Mangiante et al. [34] finds that, comparing to receiving a full 360° video through the EPC, receiving it through a MEC platform which remotely renders the requested VR frames represents a reduction of traffic in the core and the radio access of 80.5%, while maintaining an acceptable latency. This section provides a possible architecture for the delivery of such content, while addressing its challenges, based on the information found in the literature from [4, 36].

In edge computing, the video processing and storage tasks are offloaded from the core network to the edge servers. Figure 3.10 illustrates an edge computing architecture. The first stage is concerned with the capturing, stitching and encoding of the content, as described in Section 3.3. The second stage transports the videos for storage at the edge servers. The edge delivery component focuses on transporting the 360° video content from the edge servers to the clients, meeting the aforementioned requirements.

The goal of this architecture is to reduce the amount of data that has to be transported from the creation stage to the edge servers while keeping high quality and low motion-to-photon latency. The delivery of 360° videos can be made by means of distribution trees that disseminate the content from the source to the edge servers. The issue with 360° video transport is that it takes too much bandwidth to transport the entire high-quality 360° videos to all edge servers. However, viewers are likely to watch the video in spatially, temporally and/or behaviorally coherent ways. It is possible to take advantage of these coherences as follows:

- **Spatial coherence** - Since the viewport can be made of tiles that are spatially adjacent to each other, the edge server can fetch and cache tiles in advance based on the current viewport. This

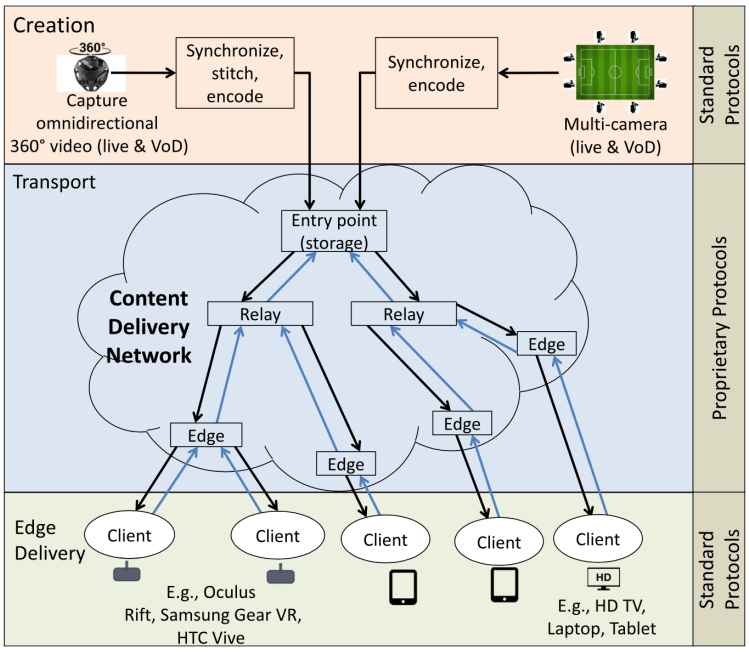


Figure 3.10: Architecture of a 360° video stream delivery system [4].

way, when the viewport changes, the required tiles may already be cached at the edge server;

- **Temporal coherence** - It is likely that the user changes its viewing direction in a predictable way over time. Knowing the trajectory of the viewport, it is possible to predict its position in the near future and prefetch the tiles that will likely be needed;
- **Behavioral coherence** - Although users have total freedom to choose their viewport, for the same video there will likely be a correlation in the viewport trajectory across different users. For example, in the case of a tennis match, most users will be likely following the ball.

If exploited correctly, all these coherences allow for less traffic throughout the network. It also allows for avoiding transporting all tiles at all qualities to the edge servers, and prefetch only what will be needed. The main challenge is to determine how to optimally deliver the right tiles in a scalable, accurate and efficient manner.

3.7 360° Video Streaming QoE

The notion of Quality of Experience (QoE) stems from the Quality of Service (QoS) metric, which assesses the overall performance of a service based on quantifiable parameters related to the network, such as bitrate, throughput, packet loss, among others.

Although QoE is largely dependent on QoS, it has a more multi-dimensional nature. Its differences to QoS have to do with covering a broader domain, which involves more than telecommunications, dealing with the user's perception of system performance instead of the performance of the system itself, and requiring a multi-disciplinary and multi-methodological approach, while QoS only relies on analytical approaches.

QoE is defined as “the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user’s personality and current state” [53]. As such, it is important to understand and measure what influences and fulfills the user’s expectations.

Understanding QoE is key to understanding how to design applications that maximize this metric and, consequently, users’ satisfaction.

This section covers the factors influencing QoE, the various methods available to assess video quality and quality assessment metrics, based on the literature in [53–59].

3.7.1 QoE Influencing Factors

Any characteristic of a user, service, system, service or application whose actual state or setting can have an influence on a user’s perceptual process, is considered a QoE influencing factor. These factors are grouped in three categories:

- **Human factors** - any variant and invariant characteristic of the user, such as demographic and socio-economic background, their physical and mental constitution, and their emotional state. The user’s visual and auditory acuity, gender, age, and others, are also QoE influencing factors;
- **System factors** - any properties and characteristics that determine the technically produced quality of an application or service. It involves the media capture, encoding, transmission network and user device, among others. The encoding influences the quality of the video in many ways, through the QP, frame rate and resolution, among others. The network may lose packets or experience delays. The users’ devices influence the experience through the screen size and resolution, battery lifetime, among others;
- **Context factors** - any factors that embrace any situational property to describe the user’s environment in terms of physical, temporal, social, economic, task and technical characteristics. The physical context refers to the location and space where the experience takes place. The temporal context covers the time of day, duration and frequency of use. The economic context includes subscription costs to services.

3.7.2 QoE Measurement Methods

It becomes clear from Subsection 3.7.1 that measuring QoE is a very subjective matter. Because of this, service providers need good QoE models in order allow them to understand how to design an application and allocate resources in order to keep users satisfied.

The existing methods can be divided into two categories, subjective and objective. Irregardless of which modelling technique is used, all QoE models provide a prediction of the perceived subjective quality of a video through some function or mapping. The metric that is normally predicted is the Mean Opinion Score (MOS), which ranges from 1 (“Bad”) to 5 (“Excellent”).

Subjective Quality Assessment Models

Subjective quality assessment consists in subjective tests performed on groups of human observers, who provide quality ratings. The objective of these tests is to understand the range of opinions the subjects express when exposed to the media content.

There is a set of recommended guidelines, issued by the Telecommunication Sector of International Telecommunication Union (ITU-T), on the standardized methodologies and test environments. These methodologies can be categorized into either double stimulus or single stimulus. In a single stimulus methodology, the subject evaluates the test without having the source as reference. In a double stimulus methodology, the subject evaluates the test in relation to the source.

The final quality score is obtained by averaging the ratings given by multiple subjects into a MOS, in the case of a single stimulus methodology, or a Difference Mean Opinion Score (DMOS), in the case of double stimulus methodology. MOS scores are largely considered to reflect the quality perceived by the Human Vision System (HVS) and therefore can also be used to validate a QoE model.

Even though subjective quality assessment experiments reliably determine perceived quality, they are time consuming, expensive, laborious and have high variance between subjects. They are also not instantaneous, meaning they cannot be incorporated into adaptive streaming systems.

Objective Quality Assessment Models

Due to the drawbacks of subjective quality assessment models, objective quality assessment models present themselves as an interesting alternative, as they can provide a real-time in-service quality assessment and replace a human test group with a computational model. The goal of these models is to predict the MOS that would be obtained with subjective experiments as closely as possible, by using objective metrics and a QoS-QoE mapping function that correlates the predicted MOS with human subjectivity.

Some models directly map an objective measurement of video quality such as the Peak-Signal-to-Noise Ratio (PSNR) metric to a prediction of user perceived quality. This approach, however, suffers from requiring a full reference comparison to the original video frames and from not taking into account the many factors that influence QoE.

Parametric QoE models derive a predicted MOS from a function of multiple objectively measured parameters, and are among the most commonly used objective quality assessment models. The used parameters may be QoS metrics such as bandwidth, delay and bit error rate but can also be related to the nature of the video stream, such as resolution, frame rate, among others. Other factors that can influence QoE and that can be considered by objective quality assessment models are:

- **Initial delay** - Initial delay is defined as the time between the user starting a streaming session and the start of the multimedia playback. This delay is caused by the amount of data that needs to be transferred from the server before decoding and playback can begin. It is also usual for the client to fill the playout buffer with an additional amount of video playtime before starting playback in order

for the client to be robust to short term throughput variations. As such, there is a trade-off between the amount of initial video playtime to fill the buffer and the initial delay and risk of buffer depletion, also known as stalling. Subjective quality assessment experiments show that most users prefer longer initial delays to experiencing sudden and unexpected stalls in the middle of the streaming session. Naturally, an objective QoE model needs to take this into account. Studies show that initial delays of up to 16 seconds only have a marginal impact on the perceived quality;

- **Stalling** - Stalling is the sudden and unexpected interruption of video playback due to insufficient video playtime in the buffer. This happens when the throughput experienced by the client is lower than the requested video bitrate. In order to recover from a stall, the client needs to fill the buffer again with sufficient video playtime to resume playback, also known as rebuffering. Again, the amount of playtime to transfer from the server while rebuffering has a trade-off with the length of the rebuffering event and the risk of incurring in a stalling event shortly after resuming playback. Subjective studies show that, although a long rebuffering event decreases perceived quality, users prefer one long stalling event to multiple short ones;
- **Rate adaptation** - As shown in Section 3.4, adaptive bitrate techniques, such as MPEG-DASH, allow the user to switch the video quality during playback in order to adapt to the available bandwidth throughout the session. Studies show that although it has a negative impact in user perceived quality, switching down the quality in a controlled manner is preferable to incurring in uncontrolled effects like stalling. Two factors that are a part of rate adaptation, and consequently can influence the quality assessment, is the mean served quality and the number of quality switches experienced throughout the streaming session.

An objective quality assessment model is usually validated by comparing its output with the results of subjective experiments where human subjects give their subjective evaluations on quality. Models applied within a network environment may also be validated by using a network simulator or a testbed.

Objective Quality Assessment Metrics

Objective measurements of video quality are important for objective quality assessment. When a video frame is compressed with a codec, it suffers loss of information. These metrics assess to which extent the used codec impairs the reference video. Most metrics used for 360° video are adaptations of metrics used for 2D video.

PSNR is a term for the ratio between the maximum possible power of a signal and the power of the corrupting noise that affects the fidelity of its representation. It is studied that the subjective sensation of a stimulus is proportional to the logarithm of its intensity. As such, PSNR is generally expressed in dB.

PSNR is most easily defined as a function of the Mean Squared Error (MSE). The MSE is defined as

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[I(i,j) - K(i,j) \right]^2 , \tag{3.1}$$

where M and N refer to the frame height and width in pixels, respectively, (i, j) refers to the location of

a pixel, I refers to the pixel value of the impaired frame at that location and K refers to the pixel value of the reference frame at that same location. To obtain a single value for the video sequence, the MSE values for each frame are averaged along all the video frames.

The PSNR is defined as

$$\mathrm{PSNR} = 10 \log_{10} \frac{p^2}{\mathrm{MSE}}, \tag{3.2}$$

where p is the peak pixel value (255 for a bit depth of 8 bits). More generally, when samples are represented with B bits, $p = 2^B - 1$.

The PSNR value approaches infinity as the MSE approaches zero. This shows that a higher image quality generates a higher PSNR. A small PSNR means that the differences between the impaired and the reference video are high. A PSNR value of 35 dB is generally considered acceptable.

In order to adapt the PSNR metric to 360° video, the adapted Viewport PSNR (V-PSNR) has been proposed. This metric applies the PSNR metric only to the pixels of a given viewport. Naturally, this metric varies for the same video, depending on the chosen viewport. As such, V-PSNR needs to be jointly presented with explicit head motion data.

The Bjøntegaard Delta PSNR (BD-PSNR) provides a method to measure the coding efficiency between two codecs which produce different pairs of bitrate and distortion. This metric is used to calculate the average PSNR difference between two sets of PSNR and bitrate measurements.

3.8 State of the Art in MEC-Assisted 360° Video Streaming

Several solutions have been recently proposed in the literature that leverage the benefits brought by edge-assisted 5G networks in order to solve some of the challenges presented in Section 3.2. This section provides a brief overview of some of these solutions, outlining their approaches and results.

3.8.1 MEC-assisted Viewport Rendering

Mangiante et al. [34] propose a viewport rendering solution at the edge of a mobile network, with the aim of reducing the bandwidth and latency required by 360° video streaming. The authors deploy a 4G LTE environment, leveraged by a MEC server, where a 360° video service produces a video stream, which is delivered to a remote server (μCloud), which in turn processes the 360° video, extracts the requested viewport, and transcodes and optimizes it for mobile devices. The computed viewport is then streamed to the user. Viewport control indications, representing the angle the user is currently looking at, are sent upstream in order for the MEC server to compute the correct viewports.

The authors refer that most VR/AR developers and industries agree that application Round Trip Time (RTT), or the time it takes between a user moving his head and the corresponding viewport showing up on the HMD display, should be less than 20 ms in order to be imperceptible to the user. This RTT is not

to be confused with network latency, since it also encompasses the latency of the sensor, the computing time taken by the device, and the time it takes for the display to refresh. The network latency must be therefore even lower than this value.

In order to assess the role and value of edge computing to reduce latency and traffic at the radio access network, the authors deploy the 4G LTE test environment illustrated by Figure 3.11.

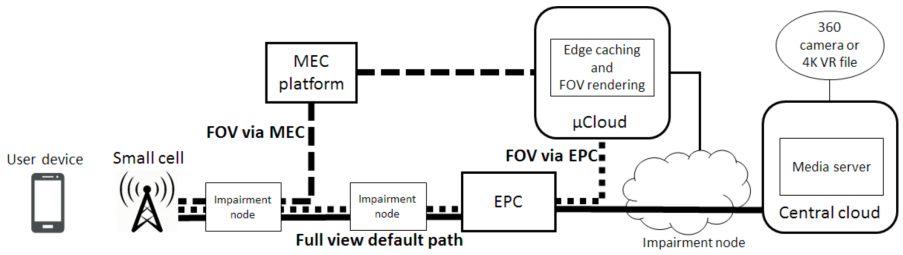


Figure 3.11: Network topology of the test environment [34].

The user device is connected to a 4G LTE base station. A MEC server is deployed between the base station and the network core. The μCloud is connected both to the MEC server and the core, in order to compare the two scenarios, represented by long-dashed and the dotted path. The user can also receive the full 360° video directly from the central cloud via the solid line.

The authors test two delivery options, represented by the long-dashed line and the solid one. The test environment is setup to have 1 Gbps available bandwidth between the central cloud and the μCloud, 22 Mbps available bandwidth between the μCloud and the user, 30 ms network latency between the central cloud and the user and 13 ms network latency between the μCloud and the user.

The authors found that, by comparing the two approaches, it is possible to achieve an 80.5% bandwidth saving. Furthermore, the full 22 Mbps bandwidth available between the μCloud and the user is not enough for the full 360° video bitrate, leading to frame drops. The rendered viewport solution only uses 5.85 Mbps.

This paper shows that it is possible to leverage MEC infrastructure in 360° video delivery as it will lead to significant benefits for all stakeholders. However, this work only considers only one device requesting only one video, which does not reflect what is likely to happen in a realistic scenario. In order for the deployment of MEC infrastructure to be worthwhile to service providers, it needs to scale well. Furthermore, the paper only compares delivering the rendered viewport via the MEC server with delivering the full video via the network core, not providing an analysis for the dotted line scenario. It would be interesting to compare the rendered viewport solution delivered both by the MEC server and the network core.

3.8.2 MEC-assisted Tiles-based Video Delivery

Hu et al. [60] propose a mobile edge assisted streaming system for 360° video in order to reduce the bandwidth and transmission delay, focusing on a live streaming scenario. This system, called MELiveOV,

offloads some of the processing tasks from the device to the edge computing enabled 5G base stations.

Figure 3.12 shows an overview of the proposed streaming system, which is composed by a 360° video generation module, a streaming module and a viewport prediction module.

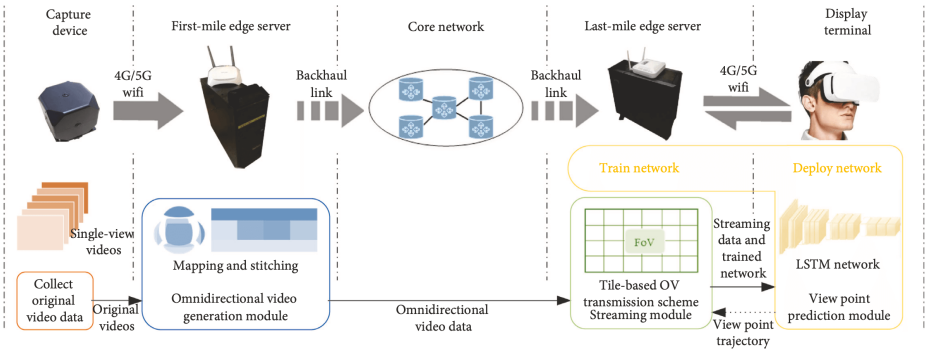


Figure 3.12: MELiveOV system overview [60].

In live streaming, the delay between the events and the delivery to the user is also a constraint. Due the time consuming nature of the on-device video generation process, this computational task is offloaded to a first-mile edge server. The mapping and stitching operations are performed by the omnidirectional video generation module.

The streaming module is deployed at a last-mile edge server. This module optimizes the transmission in real-time based on the user's viewport trajectory. The authors adopt a two-layer tile-based transmission method. The two-layer transmission method works by generating a low bitrate layer called the basic layer (BL), on top of which the tiles covered by the viewport, encoded with a high bitrate, are transmitted as a tile enhanced layer (TEL). With this method, if the TEL has not arrived in time or if the user suddenly looks in another direction, the client can display the BL to ensure a basic viewing experience instead of generating a blank area. Figure 3.13 illustrates the two-layer streaming process.

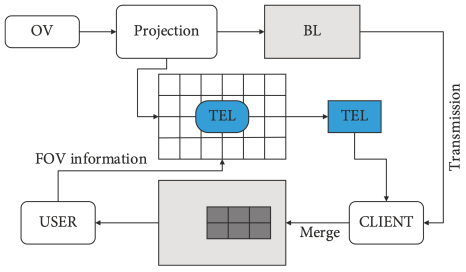


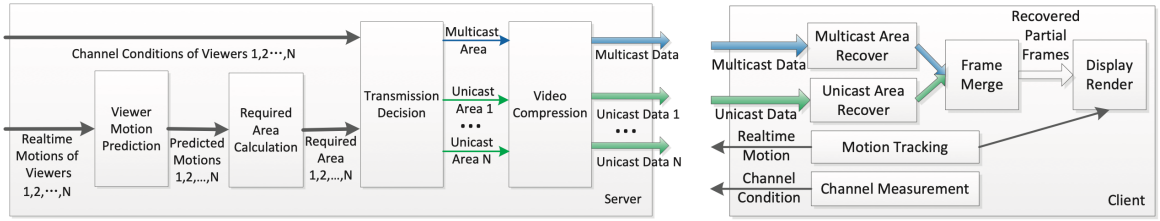
Figure 3.13: Two-layer streaming process [60].

In order to assess the performance of this system, the authors implement a prototype system consisting of an omnidirectional camera, a 5G base station, two edge servers, a virtual core network and a display terminal. By testing this prototype with several live 360° video sequences, the authors achieve a bandwidth usage reduction of about 50% comparing to an unoptimized streaming scheme, while reducing the transmission delay by about 70%. While this paper shows that MEC is a promising solution for 360° live video streaming, it also does not consider a multiuser scenario.

3.8.3 MEC-assisted Multicast

Bao et al. [61] propose a scheme to optimize the network bandwidth using motion-predicted-based multicast to serve concurrent viewers. Contrarily to what happens with 2D video, when several users are watching a 360° video together, such as a live stream of a sporting event, they can't share a single screen. Instead, each user needs their own HMD, which quickly multiplies the required bandwidth. However, taking into account that the users are watching the same video and that most follow similar motion patterns in order to focus on the areas of interest, the possibility of sending only a partial viewport to every user in a multicast manner emerges.

The authors address four main challenges: how to predict the viewer motion in the near future, how to handle prediction errors, how to transmit the data to viewers in a group in a bandwidth-efficient manner and how does partial transmission affect video compression. In order to address these challenges, they develop the framework illustrated in Figure 3.14.



(a) Server side (b) Client Side

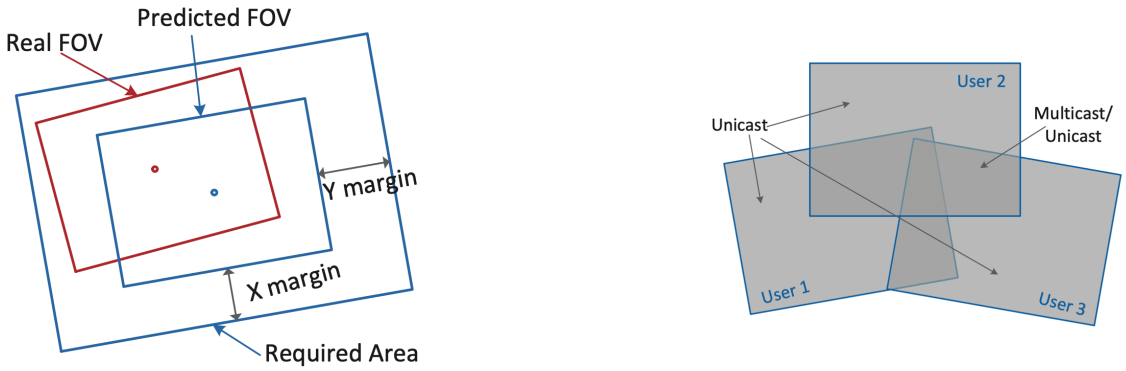
Figure 3.14: 360° video multicast system framework [61].

The server side includes modules for viewport prediction, calculating the required viewport area for each viewer, making the multicast or unicast decision for each block, based on channel conditions and performing video compression. On the client side, viewer motion and channel conditions are sent to the server. The clients receive both unicast and multicast data and join them into a single frame.

The viewport prediction is made through a neural network using present and past motions. The prediction may be done in a range of 0.1 to 0.5 seconds. Based on this prediction, the predicted viewport is obtained. Since the prediction is not entirely accurate, more than the predicted viewport needs to be sent in order to guarantee that enough pixels are received by the viewer. Figure 3.15(a) illustrates this notion. The larger the margin, the highest the probability that the user's viewport will be covered, at the expense of needing to send additional information. The authors tune the margin size by balancing this trade-off, using the pixel loss ratio as a metric. They show that by including a 15° margin both in the vertical and horizontal FoV, the pixel loss ratio is less than 0.1%.

Regarding the transmission of data in unicast or multicast, the authors consider two methods, multicast only and smart multicast to transmit the blocks. Multicast only transmits the union of all blocks to all users, regardless of the number of users that request a specific block. However, this may not be the best approach when channel conditions are diverse. In order to address this, the authors also consider a smart multicast method, which incorporates both multicast and unicast. With this method, the blocks

requested by a single user is sent in unicast only that user. For a block required by multiple users, a decision needs to be made in terms of sending it in unicast or multicast, considering the cost of either option in terms of bandwidth consumption. Figure 3.15(b) illustrates this decision.



(a) Illustration of the required area.

(b) Multicast/unicast decision.

Figure 3.15: Multicast system [61].

In the video compression module, a low latency profile that only includes I and P frames is used. Block-based compression is used, where each block consists of 8×8 or 16×16 pixels. However, because frames are partially transmitted, a block that is in a frame may not be able to reference the previous frame. Therefore, it needs to reference a past frame where that block was present. The encoding needs to happen on the fly for each frame, as different viewers have different past partial frames.

The authors evaluate the performance of the proposed framework. The performance metrics are the bandwidth consumption and the pixel loss ratio, as a function of the number of connected users. The authors assume a 0.2 second motion prediction window.

First, a group of users with similar channel conditions is assumed. Without multicast, the bandwidth consumption grows linearly with the number of users. Compared with full multicast, the smart multicast and multicast only methods achieve more than 50% bandwidth saving. More interestingly, the bandwidth consumption grows much slower with each additional user.

In terms of the pixel loss ratio, its value gets smaller with more viewers. This is due to the pixels from the multicast area being received by all users. With more users, this area will be larger, and therefore there will be more room for the real viewport to deviate from the predicted viewport. The pixel loss ratio of the multicast only mode decreases faster than the smart multicast mode, because more data is being shared among users.

This paper shows that there is are clear bandwidth savings when taking advantage of the multicast opportunities that 360° video provides. However, this solution only works when the users are watching the same video simultaneously, making it a good fit for live streaming situations. However, for on-demand content, where each user may be watching a different video, or be watching the same video but at different times, the multicasting opportunities become much smaller.

Chapter 4

Simulator Implementation

This chapter describes the implementation of the simulator. The simulator's concept is explained in Section 4.1. Section 4.2 explains how the throughput is calculated, while Section 4.3 describes the implementation of the rate adaptation algorithm. Section 4.4 explains how the video bitrates were estimated. Section 4.5 describes the implementation of the scheduling algorithm. The way the latency affects the scheduling of resources is explained in Section 4.6. Section 4.7 describes how the viewport trajectory affects video quality and Section 4.8 describes the used QoE model. Finally, Section 4.9 presents a complete flowchart of the simulator.

4.1 Introduction

In order to assess and compare the viability and performances of the different 360° video streaming schemes discussed in Section 3.5, a previously developed simulator for LTE networks and 2D video was used as a starting point for the development of a new network simulator in MATLAB[1] [62]. The idea of this simulator is to replicate the state of a 5G network during a 360° video streaming session of a certain duration, with a certain number of connected users, and in which a certain streaming scheme is employed. The simulator then calculates the experienced conditions by each user throughout the session, and inputs this information into a QoE model. The users' satisfaction computed by this model is then used as a metric to evaluate the performance of each method.

Another objective of this simulator is to analyze the performance gains enabled by deploying MEC server infrastructure on the edge of the network. Figure 4.1 illustrates the implemented network topology. When a MEC server is not deployed, the client needs to fetch the requested segments from the central server, through the core network. With a MEC server, assuming the relevant segments are cached and/or assuming no backhaul limitations, the client can fetch the requested segments from the edge of the network, resulting in much smaller end-to-end latencies.

The simulator was developed in a modular form, in order to allow the reconfiguration of the various blocks and the implementation of different algorithms. The general workflow is presented in Figure 4.2.

First, the simulation settings are imported. The main settings include the total simulation time, which is equivalent to the length of the streaming session to be simulated, the number of connected users, which delivery scenario is being simulated, the 5G numerology configuration, among others.

[1] Simulator code available: `https://github.com/miguel-loff/360_streaming_simulator`

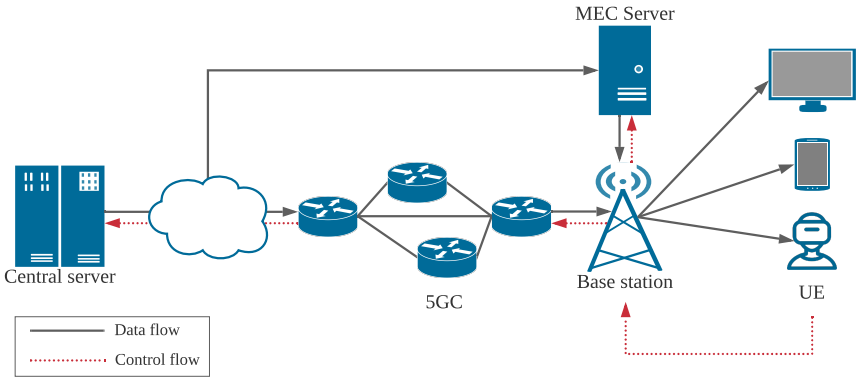


Figure 4.1: Network topology.

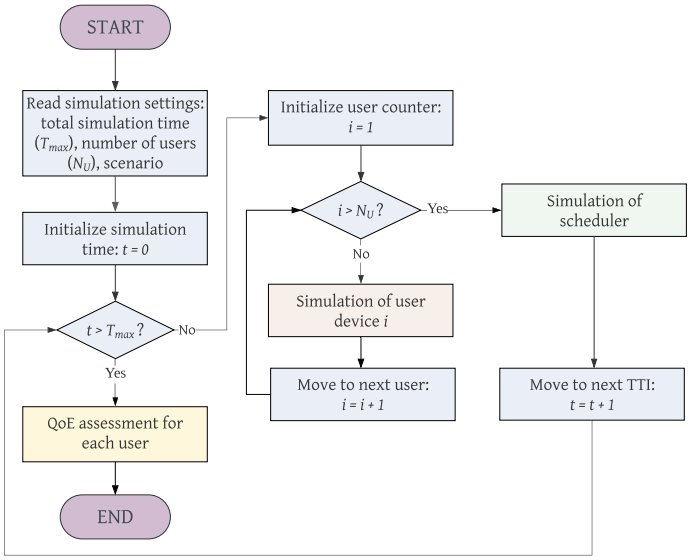


Figure 4.2: General simulator workflow.

After the settings are imported, the simulator enters its main *for* loop. The number of iterations of this loop is equivalent to the number of TTIs one wishes to simulate, which is given by the total simulation time and the duration of each TTI, which depends on the used numerology. For each iteration, the state of each user's device is simulated: the viewport orientation is updated, the buffer level is updated, the rate adaptation algorithm may request new segments to the server and select their respective quality, and the CQI may be reported to the base station.

Following the simulation of all users' devices, the resource scheduler is simulated. In each TTI, the scheduling algorithm distributes the available PRBs among the users that need to receive data. Different algorithms can be used depending on the desired behavior, and are generally based on the users' performance metrics. When the simulator reaches the predefined simulation time, the QoE is assessed for each user.

The following sections describe in more detail the components of the simulator.

4.2 User Throughput

Although the simulator does not consider packet loss during the streaming session, it assumes that the channel quality experienced by the users varies throughout the simulation time. The channel quality, which is estimated by the User Equipment (UE) and reported to the base station by means of the CQI, defines how many bits can be transmitted with each PRB.

To calculate the instantaneous throughput per PRB, (2.1) is used with the number of PRBs, $N_{PRB}^{BW(j),\mu}$, set to 1. It is also assumed that there is no carrier aggregation, the scaling factor is 1, and that the overhead value is set to 0.14 since the considered frequency range is FR1. The number of layers depend on the used MIMO configuration.

The modulation order and coding rate depend on the reported CQI. A higher CQI will allow the base station to use a higher modulation order and code rate, resulting in a higher throughput. The values are used according to Table 2.4. A poor channel is assumed instead of a normal channel because, in an initial phase of 5G, devices may not support a 256-QAM modulation scheme.

Table 4.1 presents the number of bits per PRB for each CQI level, assuming a 2x2 MIMO configuration, and the corresponding throughput assuming 106 PRBs per TTI.

Table 4.1: Instant throughput for each CQI level.

CQI	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Throughput [bits/PRB]	44	68	109	174	253	340	427	553	695	789	960	1128	1307	1478	1605
Throughput [Mbps]	4.7	7.2	11.5	18.4	26.9	36	45.2	58.6	73.7	83.6	101.8	119.5	138.6	156.7	170.1

All the users report their CQIs to the base station every 5 ms. The used CQI value dataset was taken from [62]. The author obtained the values through a simulation made using OMNeT++, an open-source discrete event simulator used for simulating networks, with the INET-Framework extension to simulate network communication with the TCP/IP protocol. SimuLTE, which is a tool enabling system level performance evaluation of wireless networks, was also used.

A scenario was simulated containing a single base station and 200 mobile users randomly roaming around the base station. The reported CQIs by each user were recorded along a 3 minute simulation. The obtained results have enough heterogeneity so that some users always constantly report a CQI of 15 and others fluctuate around low and high values. However, the majority of the users report a good CQI most of the time.

4.3 Rate Adaptation Algorithm

The server stores the videos, divided in segments of equal time and with several quality levels. In the initial buffering phase, the users request a set number of segments of a set quality level. However, after this stage, in order for MPEG-DASH to work, there needs to be an algorithm that selects the quality of the subsequent segments according to the available bandwidth. The behavior of this algorithm is not standardized by MPEG-DASH and therefore the client has the autonomy to implement its own rate

adaptation algorithm. There are several options available. Some algorithms prioritize not depleting the buffer at all costs, while others prioritize maintaining a stable quality level throughout the session. The client also needs to determine when to request a new segment (i.e., when to call the rate adaptation algorithm). This decision is not part of the algorithm and it is also up to the client to define the ideal strategy.

It is quite clear that the simulator should not call the rate adaptation algorithm before a previously requested segment has finished downloading as it would cause conflicts of information. Therefore, the first general rule is that the rate adaptation algorithm should only be called if the client is not currently requesting any segment.

There is also the question of buffer overflow. Since the client's buffer capacity is not infinite, it does not make sense to request a new segment at a time when it would exceed the buffer's limit, which would lead to a wastage of resources. Therefore, the second rule is that the rate adaptation algorithm should only be called to request the next segment when the client's buffer falls below a certain threshold, α.

In addition to this, it is essential to consider that the timing in which the next segment request is made directly impacts the streaming session quality. One possibility is to set the threshold α to a higher buffer level, which means the client would request the next segment sooner in relation to the instant the current segment finishes playing. This way, the client would have more time to download the new segment before the buffer is depleted, reducing the risk of incurring in an undesirable playback stall. Therefore, it seems that the best strategy is to set the threshold α to a value as high as the buffer capacity allows. While this is certainly true for a viewport-independent streaming scheme, when dealing with a viewport-dependent streaming scheme such as the tiles, partial tiles or viewport only schemes described in Subsections 3.5.3 and 3.5.4, it is important to remember that certain parts of the viewing sphere outside of the viewport are delivered in a lower quality or not sent at all, depending on the specific scheme. The only factor that goes into deciding which tiles or viewports the server should send is the requested direction sent by the client along with the next segment request. Since the users move their head during the time between the segment request and the segment playback, this means that requesting the segment sooner will result in a higher mismatch between the requested direction and the actual direction the user is viewing when the segment is played back, up to the point where the delivered viewport could be completely outside of the actual viewport. Therefore, the threshold α presents a trade-off between the risk of stalling and the risk of delivering an inaccurate viewing direction. Naturally, the ideal value for α is different for each streaming scheme, as some are more tolerant than others in terms of the mismatch in viewing directions, therefore allowing for a higher threshold. A sensitivity analysis of the ideal value for α is presented in Section 5.3 for each streaming scheme, so that they can be fairly compared with each other.

The implemented algorithm is an adaptation of the QoE-enhanced Adaptation Algorithm over DASH (QAAD), presented in [63]. This algorithm takes the clients' current buffer level and the experienced throughput into account in order to always try to preserve a minimum buffer level to avoid stalling and to minimize the number of quality changes during the playback, which is shown to enhance the QoE. The rate adaptation algorithm consists of two parts: the throughput estimation and the bitrate selection.

4.3.1 Throughput Estimation

A segment-based scheme, such as the one used in [64], was selected. Using this scheme, the estimated channel throughput is updated every time a new segment finishes downloading as

$$bw_{sample} = \frac{K}{T}, \tag{4.1}$$

where K is the size in bits of the downloaded segment and T is the time it took to download it, from the moment it was requested to the moment it finished downloading. This scheme, while also showing a smooth variation of the estimated bandwidth, which is desirable for multimedia streaming, may suffer from taking more time to reflect recent trends in the available throughput variation, as the update frequency is dependent on a segment finishing being received. This may cause a delayed adaptation behavior. A weighted moving average scheme, such as the one is used. A weight factor of $\omega = 0.3$ has shown to most closely resemble the behavior of the originally used scheme.

4.3.2 Bitrate Selection

One of the objectives of QAAD is to minimize the number of quality changes. The bitrate selection scheme takes into account the current buffer level of the client as well as the available throughput. Each time the algorithm is called, it starts by determining the best quality it is possible to achieve, l_{best}, by taking into account the highest bitrate it is possible to request without surpassing the estimated available bandwidth, $bw_{estimated}$. The used algorithm is presented in Algorithm 1, which is a modification from the original described in [63]. The algorithm, its modifications and validation tests are described in Annex A.

Algorithm 1 Modified pseudocode of the bitrate selection scheme in QAAD [63].

if $l_{best} == l_{prev}$ **then**
    $l_{next} = l_{prev}$
else if $l_{best} > l_{prev}$ **then**
    if $B > \mu$ **then**
        $l_{next} = l_{prev} + 1$
    else
        $l_{next} = l_{prev}$
    end if
else if $l_{best} < l_{prev}$ **then**
    $k = 0$
    do
        $t_{l_{prev}-k,\sigma} = \frac{B(t)-\sigma}{1-\frac{bw_{estimated}}{b(l_{prev}-k)}}$
        if $t_{l_{prev}-k,\sigma} < 0$ **and** $B\,(t) - \sigma > 0$ **then** ▷ Modification
            $k = k + 1$ ▷ Modification
            break ▷ Modification
        end if ▷ Modification
        $n_{l_{prev}-k} = \frac{t_{l_{prev}-k,\sigma} \cdot bw_{estimated}}{\tau \cdot b(l_{prev}-k)}$
        $k = k + 1$
    while $n_{l_{prev}-k} < 1$ **and** $k < l_{prev}$ ▷ Modification
    $l_{next} = l_{prev} - (k-1)$ ▷ Modification
end if

4.4 Video Bitrates

In order for the simulator to fulfill the objective of realistically simulating the behavior of a 5G network in a 360° video streaming scenario, real bitrates need to be considered for the available video quality levels. Furthermore, in order for the simulator to also fulfill the objective of comparing different 360° streaming delivery schemes, the bitrates for the various schemes need to be comparable, as it is important that the video quality yielded by a given quality level in one scheme is comparable to the one yielded by the same level in another scheme. This section details how those values were obtained, based on the work in [52]. The following delivery schemes were studied:

- **Monolithic Equirectangular (MonoEqui)** - It is based on the viewport-independent streaming scheme presented in Subsection 3.5.1, using the equirectangular projection. The entire frame is encoded with the original resolution and with some target QP. The target QP is varied to obtain different rate-distortion points and thus obtain different target quality levels and bitrates;
- **Tiles OMAF Spatial Resolution (OMAF-SRes)** - It is based on the tiles-based streaming scheme presented in Subsection 3.5.3, using the equirectangular projection. Tiles are encoded with a high and low spatial resolution. The selection of the high spatial resolution tiles is made based on the coverage of the viewport. The low spatial resolution tiles have half the resolution of the high resolution tiles, while the QP is maintained for the same quality level. In addition to this, the tiles that cover the poles are encoded with half the spatial resolution comparing to the tiles of the equator. This means that a low resolution pole tile will have a quarter of the spatial resolution of the original tile. The different target quality levels are obtained by changing the resolution of all tiles, while keeping the resolution ratio between high and low resolution tiles;
- **Tiles OMAF Spatial Resolution Partial Delivery (OMAF-SRes-Partial)** - This strategy is the same as OMAF-SRes, however only the tiles covering the viewport (i.e., the high spatial resolution tiles) are streamed;
- **Viewport Only (Viewport-Only)** - It is based on the viewport-only streaming scheme presented in Subsection 3.5.4. It is assumed that the server keeps the original equirectangular video and encodes the requested viewports on demand. The transmitted frame has the resolution of the viewport and is encoded with some target QP in order to obtain different target quality levels.

4.4.1 Fixed Viewport Evaluation

In [52], the author studies the rate-distortion curves of the MonoEqui, OMAF-SRes and OMAF-SRes-Partial schemes considering fixed viewports for three test sequences selected from the JVET dataset [65]. The test sequences are called *ChairliftRide*, *SkateboardInLot* and *KiteFlite*. *ChairliftRide* is reported to have moderate motion content, *SkateboardInLot* to have fast motion content and *KiteFlite* to have low motion but containing high spatial detail. All videos use the equirectangular projection, with a spatial resolution of 7680×3840 pixels, a temporal resolution of 30 frames/s and a duration of 10 seconds. Figure 4.3 presents the first frame of each video sequence.

(a) *ChairliftRide* (b) *SkateboardInLot* (c) *KiteFlite*

Figure 4.3: First frame of the the test video sequences [52].

The spatial resolution of the video is maintained at 7680×3840 for the MonoEqui. For OMAF-SRes and OMAF-SRes-Partial, the spatial resolutions associated with each Rate-Distortion (RD) point are presented in Table 4.2.

Table 4.2: Spatial resolutions for OMAF-SRes and OMAF-SRes-Partial [52].

RD Point	HSR Equator	LSR Equator / HSR Pole	LSR Pole
1	7680×3840	3840×1920	1920×960
2	6144×3072	3072×1536	1536×768
3	4608×2304	2304×1152	1152×576
4	3072×1536	1536×768	768×384

Each of the test sequences was encoded with an HEVC reference software, using a Random Access Profile (RAP) with a GOP size of 16 frames. For each video, the QP was adjusted to match the bitrate of each RD point as closely as possibly to the bitrates with 6 Mbps, 4 Mbps, 2.5 Mbps and 1 Mbps for RD points 1, 2, 3 and 4, respectively. Table 4.3 presents the QP used for each video sequence and delivery scheme, where videos 1, 2 and 3 are the *ChairliftRide*, *SkateboardInLot* and *KiteFLite* test sequences, respectively.

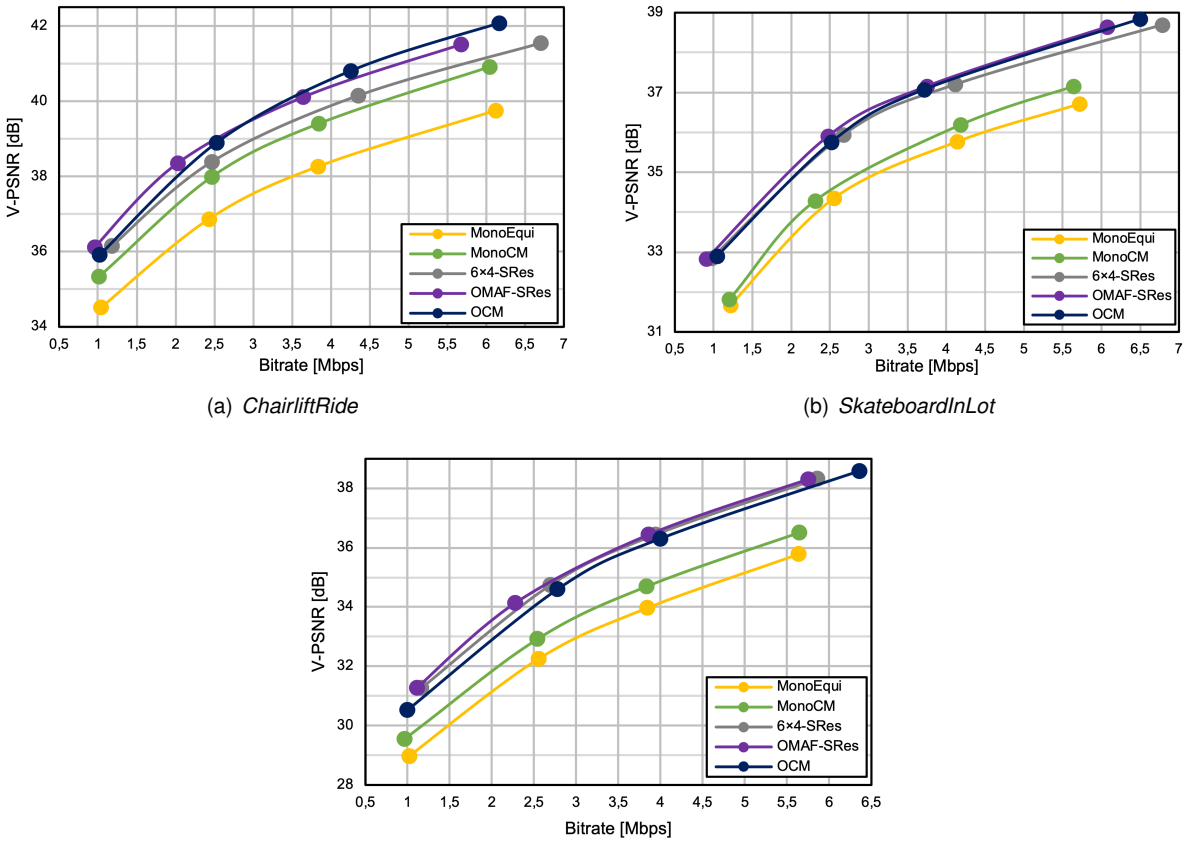
For each strategy, the viewports were rendered for each sequence for a static viewing direction of (0° , 0°). The viewport dimensions were 2000×2000 pixels, which roughly equates to a 96° vertical and horizontal FoV. An ideal streaming scenario was considered, meaning that, for tiles-based schemes, only the tiles covering the viewport had high resolution.

Each rendered viewport was then compared to the same viewport rendered from the original video by calculating the V-PSNR metric described in Section 3.7.2 for each frame and averaging all values. As

Table 4.3: QP values used for fixed viewport evaluation [52].

RD Point	MonoEqui			OMAF-SRes / Partial		
Video	1	2	3	1	2	3
1	34	36	37	30	32	33
2	37	38	40	31	33	34
3	40	41	43	32	33	35
4	46	51	51	33	36	36

such, only what is seen inside of the viewport counts toward the V-PSNR metric, but all of the transmitted video counts toward the respective bitrate. Figure 4.4 presents the rate-distortion curves for each video sequence and delivery scheme. From these curves, it is possible to observe the differences between the various delivery schemes. Naturally, the MonoEqui scheme is the one that achieves lower V-PSNR values for the same quality levels, since it is the most demanding scheme in terms of bitrate. The OMAF-SRes achieves one of the best performances.



(a) *ChairliftRide* (b) *SkateboardInLot*

(c) *KiteFlite*

Figure 4.4: RD curves for different delivery schemes and video sequences. Fixed viewport with direction (0° , 0°) [52].

4.4.2 Bitrates Estimation

The RD curves presented in Figure 4.4 illustrate the relationships between the delivery schemes, making it possible to use the corresponding data to calculate the bitrates of the quality levels for the various schemes that should be made available for the client to request. It is also possible to observe in these graphics that the author did not include the RD curves for the tiles-based partial and viewport-only rendering schemes, as it was deemed that wireless networks at the time of writing were not developed enough to support these schemes.

However, for the OMAF-SRes-Partial scheme, the author [52] calculated the BD-PSNR metric for the three video sequences. The BD-PSNR is a metric that corresponds to the average V-PSNR difference between the RD curves of two schemes, given in dB. The BD-PSNR for the OMAF-SRes-Partial scheme

is 3.13 dB, 2.55 dB and 3.78 dB for the *ChairliftRide*, *SkateboardInLot* and *KiteFlite* sequences, respectively. This means that by adding these values to the V-PSNR values of the MonoEqui RD curve, it is possible to obtain the OMAF-SRes-Partial RD curve.

For the Viewport-Only case, the bitrate values had to be estimated with the knowledge from the way this scheme is implemented. First, from the point of view of the V-PSNR, it is clear that for each RD point, this value is the same as in MonoEqui, since this metric measures the PSNR inside the viewport and the viewport of both schemes is exactly the same. What changes is the bitrate of each RD point, since only the video inside of the viewport is delivered. It is known that the original video has a spatial resolution of 7680×3840 pixels and the rendered viewport has a resolution of 2000×2000 pixels, meaning that, theoretically, the bitrate for the Viewport-Only scheme could be as low as 13.6% of the bitrate of the MonoEqui scheme. However, as described in Subsection 3.5.4, it is necessary to take into account the fact that, while the MonoEqui scheme uses a Random Access Profile (RAP) to encode the video, the Viewport-Only scheme uses a Low Delay P Profile (LDPP), since it cannot encode B frames, which will lead to slightly higher bitrates. In [66], the average bitrates for some QP values were studied for the various encoding profiles of HEVC. From these values, it is possible to calculate the loss of compression efficiency between LDPP and RAP for each of the provided QPs, and then perform a regression on these values in order to estimate the loss of compression efficiency for the QPs in the used video sequences. Figure 4.5 illustrates this regression. Therefore, in order to estimate the bitrate associated to each RD point in the Viewport-Only scheme, the corresponding bitrate in MonoEqui is multiplied by 13.6% and then by the compression efficiency ratio for the corresponding QP.

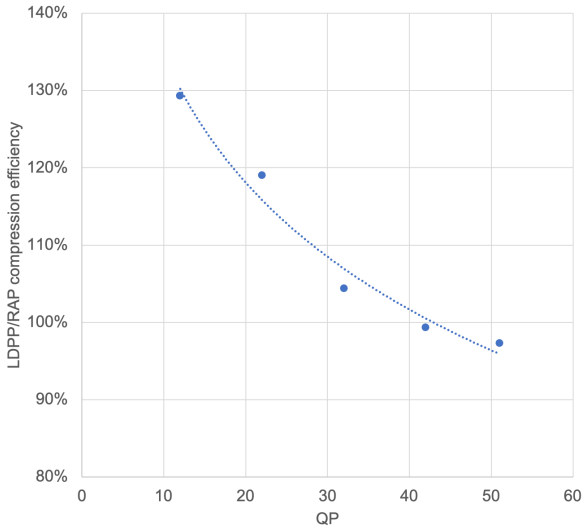


Figure 4.5: Ratio of the LDPP/RAP compression efficiency.

To calculate the bitrates of the quality levels for the various schemes, a power regression was applied to the RD points from Figure 4.4. This type of regression was chosen as it is the one the better approaches a real RD curve, since the V-PSNR tends to $-\infty$ as the bitrates tends to zero and starts to grow slower as the bitrate increases. Figure 4.6 illustrates the regressions for the three video sequences.

For reasons explained in Section 4.8, a decision was made to use 7 different quality levels. It is desirable

that, between any two quality levels, the increase in video quality is of the same magnitude. Since the V-PSNR can be used as an objective quality assessment metric, for each of the video sequences, seven V-PSNR values were linearly spaced between the maximum and minimum V-PSNR values of the MonoEqui scheme (i.e., the values associated to the 1 Mbps and 6 Mbps bitrates). Then, using the previously obtained regression equations for each of the delivery schemes and video sequences, the bitrates associated to those V-PSNR values were calculated for each scheme. This way a certain quality level in one scheme has the same V-PSNR comparing to the same quality level in another scheme, with the bitrate differing. These resulting bitrates are presented in Tables 4.4, 4.5 and 4.6.

Table 4.4: Estimated bitrates for each quality level and delivery scheme. Sequence: *ChairliftRide*.

Quality level	**1**	**2**	**3**	**4**	**5**	**6**	**7**
V-PSNR [dB]	34.5	35.3	36.2	37.1	37.9	38.8	39.7
MonoEqui [Mbps]	1.000	1.372	1.869	2.528	3.394	4.527	6.000
OMAF-SRes [Mbps]	0.564	0.764	1.028	1.373	1.822	2.402	3.148
OMAF-SRes-Partial [Mbps]	0.299	0.422	0.591	0.820	1.129	1.543	2.094
Viewport-Only [Mbps]	0.134	0.186	0.256	0.350	0.476	0.642	0.860

Table 4.5: Estimated bitrates for each quality level and delivery scheme. Sequence: *SkateboardInLot*.

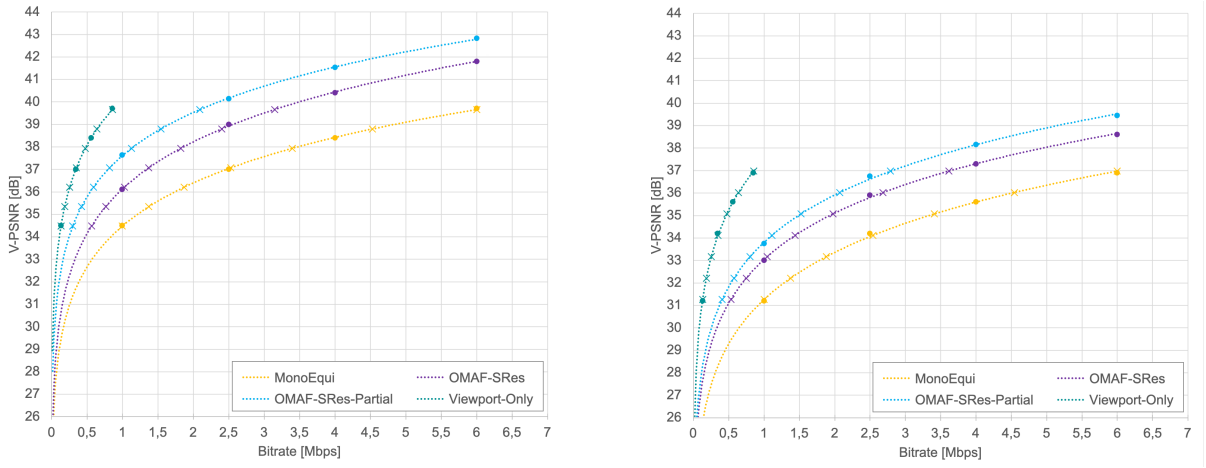
Quality level	**1**	**2**	**3**	**4**	**5**	**6**	**7**
V-PSNR [dB]	31.3	32.2	33.2	34.1	35.1	36.0	37.0
MonoEqui [Mbps]	1.000	1.378	1.880	2.543	3.412	4.542	6.000
OMAF-SRes [Mbps]	0.563	0.796	1.113	1.541	2.116	2.880	3.889
OMAF-SRes-Partial [Mbps]	0.407	0.574	0.802	1.110	1.523	2.070	2.790
Viewport-Only [Mbps]	0.131	0.183	0.253	0.348	0.473	0.638	0.854

Table 4.6: Estimated bitrates for each quality level and delivery scheme. Sequence: *KiteFlite*.

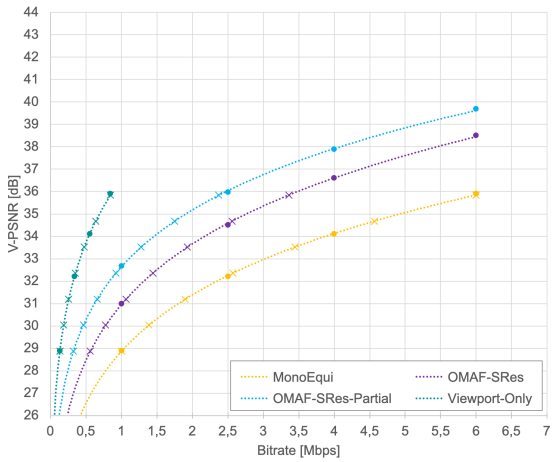
Quality level	**1**	**2**	**3**	**4**	**5**	**6**	**7**
V-PSNR [dB]	28.9	30.0	31.2	32.4	33.5	34.7	35.8
MonoEqui [Mbps]	1.000	1.387	1.899	2.571	3.444	4.567	6.000
OMAF-SRes [Mbps]	0.561	0.777	1.065	1.441	1.929	2.558	3.360
OMAF-SRes-Partial [Mbps]	0.321	0.462	0.657	0.921	1.277	1.750	2.373
Viewport-Only [Mbps]	0.130	0.183	0.255	0.349	0.473	0.635	0.845

4.5 Scheduling Algorithm

After the rate adaptation algorithm makes the decision to fetch a new segment of a certain quality level, the client makes this request to the server. As seen in Figure 4.1, the request travels through the core and into the central server, or into the MEC server. The server, in turn, sends the requested data. When the data arrives from the server, the base station needs to decide how to best distribute the available resources (i.e., PRBs) among all the clients that are requesting data. It does so by means of a scheduling algorithm. Different schedulers employ different strategies to try to serve the clients according to their



(a) *ChairliftRide* (b) *SkateboardInLot*



(c) *KiteFlite*

Figure 4.6: Regressed curves for different delivery schemes and estimated bitrates, fixed viewport with direction (0° , 0°) [52].

expectations. For example, some schedulers may try to maximize the efficient use of the resources, while others may try to provide a certain minimum QoE to all users.

The scheduling algorithm is implemented in a way that, independently from the 5G network configuration in terms of SCS, it makes allocation decisions every TTI, as this is the basic time unit of the simulator.

Resource allocation for each user is based on the comparison of a per-PRB metric [67]. The k-th RB is allocated to the j-th user if its metric $m_{j,k}$ is the largest out of all the users, i.e., if it satisfies

$$m_{j,k} = \max_i \{m_{i,k}\}. \tag{4.2}$$

Every TTI, the scheduling algorithm performs the resource allocation for the next TTI. As such, the simulation of user device i starts with updating the buffer level with the video received through the PRBs allocated in the previous TTI.

After comparing the behavior of several scheduling algorithms, the decision was made to implement the

Proportional Fair (PF) algorithm [68]. This algorithm falls under a category known as channel-aware schedulers [69]. This type of scheduler takes advantage of the periodically sent CQI to estimate the maximum achievable throughput for the i-th user, t-th TTI and k-th RB, i.e. $r_k^i(t)$.

The PF algorithm provides a good compromise between throughput and fairness among users. While it tries to maximize the total throughput, at the same time it tries to provide a minimum QoS to all users. The metric is defined as

$$m_{i,k}^{PF} = \frac{r_k^i(t)}{r^{(avg)} + \sum_{k=1}^{K} \delta_{ik} r_k^i(t)}, \tag{4.3}$$

where $r^{(avg)}$ is the averaged throughput experienced by the user up to the current TTI. The sum over all K PRBs in the denominator accounts for PRBs that have already been allocated to user i in the current TTI. δ_{ik} equals one if user i has been allocated PRB k, and equals zero otherwise. The effect of this is that, when one PRB is allocated to a user, the metric decreases, giving other users a higher chance to be served. With the PF scheduling algorithm, even users with bad channel conditions are surely served within a certain amount of time. The implementation and validation tests are presented in Annex B.

4.6 Latency

In order for the simulator to better reflect what happens in a real scenario, it is necessary to take into account the end-to-end latency that exists between the client requesting data and that data arriving at the base station. The end-to-end latency is incorporated into the simulator as a definable, fixed value, L. When a user makes a request, the scheduling algorithm waits L TTIs before it starts including that user into the valid users array in line 1 of Algorithm 4. The scheduling algorithm interprets a valid user as a user that is requesting data and made that request at least L TTIs ago.

The impact of this is that the larger the value of the latency, the more time users will need to wait before being served. This increases the risk of the client depleting the buffer and, in the case of partial delivery scenarios such as OMAF-SRes-Partial or Viewport-Only, makes it necessary to request new segments earlier in order to keep the buffer from depleting, increasing the risk of a high viewing direction mismatch because the user moved their head too much in the time between making the request and playing the corresponding segment.

As discussed in Subsection 2.5.3, the amount of end-to-end latency present is impacted by several factors and depends heavily on the implemented network. The values presented in Table 2.6 refer to a best-case scenario, i.e., to what 5G networks are expected to achieve in the future in order to fulfill the objectives defined by IMT-2020.

In reality, these values rely on the implementation of technologies that are not yet present in current 5G networks, such as the Standalone (SA) mode described in Subsection **??**, the use of network slicing, described in Subsection 2.3.3, and the deployment of a MEC server infrastructure. Even with these technologies in place, it is still possible that achieving single-digit latency values in real-world scenarios

may only be achieved on private 5G networks [70].

In order to fulfill one of the simulator's stated objectives of analyzing the performance gains enabled by deploying MEC server infrastructure on the edge of the network, it is important to estimate realistic latency values, since the use of this technology heavily influences this metric. This estimation is made based on [71].

As described in Subsection 2.3.2 and illustrated in Figure 2.6, the architecture of the NG-RAN can be implemented and deployed in different ways, depending on the network requirements. The NG-RAN consists of a set of base stations, known as gNBs, which are connected to the 5G Core Network (5GC) and to each other. The gNB incorporates the CU, DU and RU. As what happens in classical networks, the base station can be deployed as a monolithic unit incorporating all three components. Alternatively, it can be split between the CU, DU and RU. CUs, DUs and RUs can be deployed at locations such as cell sites, transport aggregation sites and edge sites.

The choice of where to implement the splits is a critical decision in terms of the resulting end-to-end latency. For example, in an ultra-low latency scenario, it would make sense to have the DU/CU close to the RU.

Figure 4.7 shows the typical latencies associated with each of the splits. As it is possible to observe in this figure, the backhaul has a high level of associated latency. Fundamentally, this is why the deployment of MEC infrastructure is essential for achieving ultra-low latency. Since MEC servers are deployed in the fronthaul, fetching data from them instead of from servers connected to the core network has the potential of eliminating the 10 ms component from the end-to-end latency.

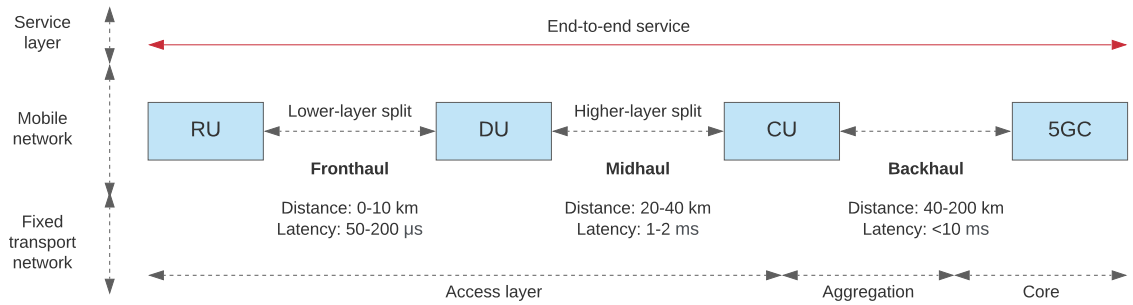
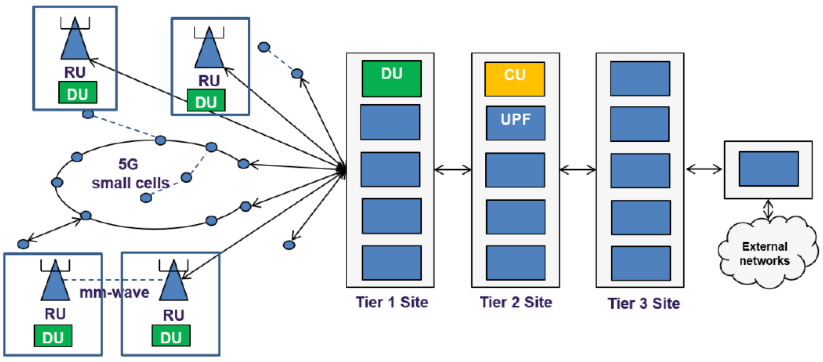


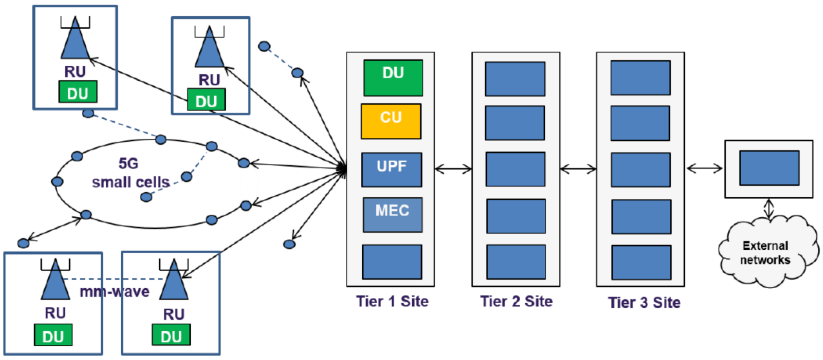
Figure 4.7: NG-RAN typical latencies (adapted from [71]).

Figure 4.8 shows different deployment topologies concepts that may be deployed to fulfill different latency requirement scenarios. Figure 4.8(a) shows a service with tolerance for a 10 ms end-to-end latency. In this case, the CU is deployed at a Tier 2 site (e.g., a central office), while the DU is deployed at a Tier 1 aggregation site or to the cell site itself. In Figure 4.8(b), MEC servers are deployed at the edge facility in order to host and run latency-sensitive application and achieve sub-5 ms end-to-end latencies. In Figure 4.8(c), the CU and DU are co-located with the RU, or even integrated into a single unit, at the cell site. Because in this scenario ultra-low latency is required, the MEC servers are also deployed at the cell site in order for applications to run locally.

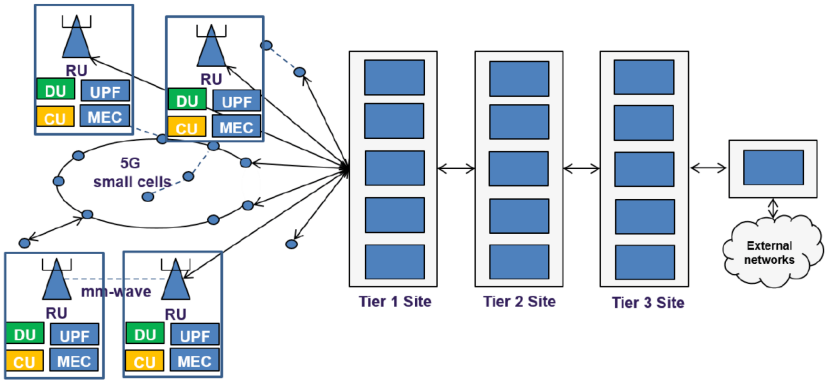
In the context of the simulator, in order to study the impact of the presence of MEC infrastructure on the



(a) <10 ms latency scenario



(b) <5 ms latency scenario



(c) 1 ms latency scenario

Figure 4.8: Deployment topologies for different latency scenarios [71].

users' QoE, the latency value, L, is set accordingly. It was decided to use the latency values presented in Table 4.7, depending on whether the existence MEC servers is being considered or not.

Table 4.7: User latency values adopted for the simulation.

MEC servers	Deployed	Not deployed
Latency [ms]	1	10

4.7 Viewport Trajectory

Except in the case of the MonoEqui scheme, where the video quality distribution across the frame is uniform, the video that is sent from the server to the client has an area where the quality is higher. When the client makes the request for a new segment, it sends, along with this request, information on the user's viewport orientation so that the server can make the decision on which tiles to send in higher or lower quality in the OMAF-SRes scenario, which tiles to send and not to send in the OMAF-SRes-Partial scenario, and what exact viewport to send in the Viewport-Only scenario.

Assuming that no viewport prediction algorithm is being used, the server can only rely on the information about the viewport orientation at the time of the request to make this decision. This means that, due to the latency and to previously buffered video that still hasn't been played back, by the time the user sees that segment, they may have moved their head from the orientation it was in at the time of the request, causing the actual viewport and the requested viewport to be misaligned.

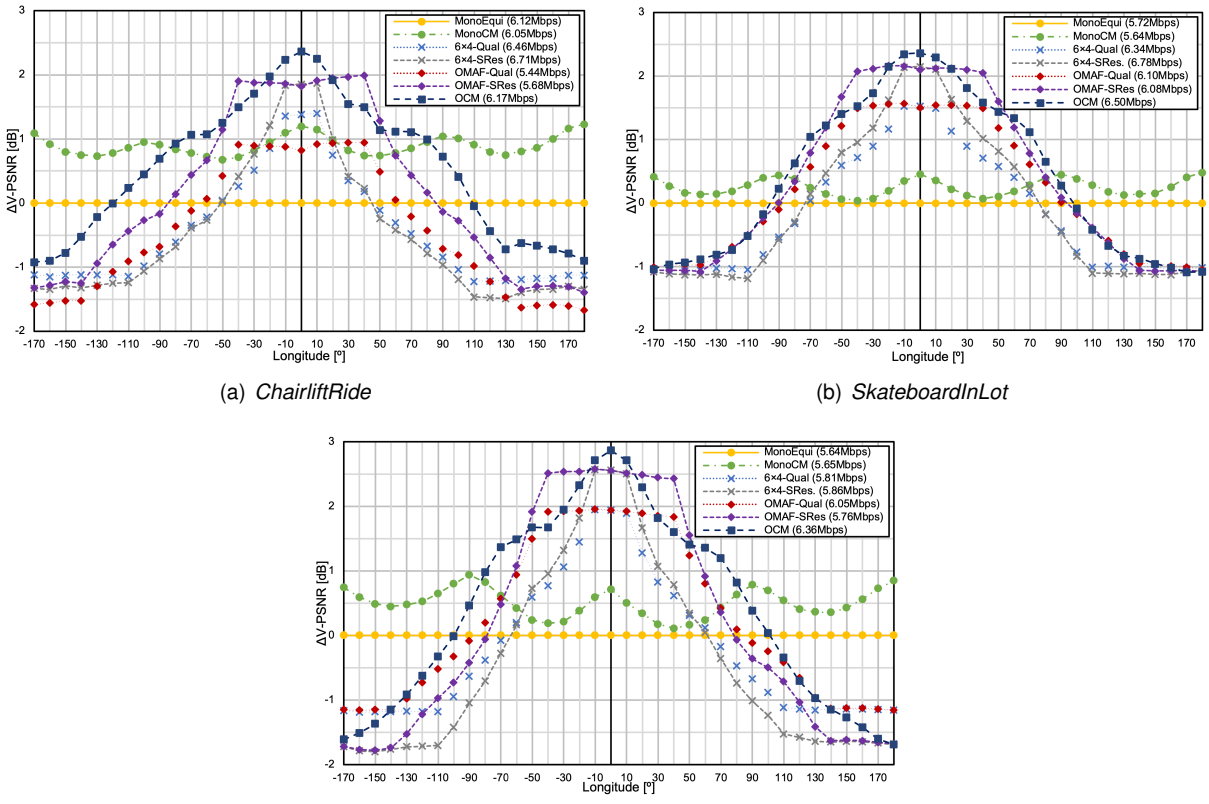
It should be noted that, in the case of the OMAF-SRes and OMAF-SRes-Partial schemes, since the video is spatially segmented into a finite number of tiles, the center of the delivered tiles' longitude cannot exactly match the requested angle. Assuming that 4 tiles are always delivered, analyzing Figure 3.9(a), their center longitued can be: $\{$-180° , -135° , -90° , -45° , 0° , 45° , 90° , 135° , 180° $\}$.

4.7.1 Fixed Viewport Evaluation with Mismatch

Naturally, the impact of this misalignment on the experienced video quality depends on the used delivery scheme. In the case of OMAF-SRes, if the viewport strays from the area of the high quality tiles, the user will see the tiles with lower quality. In the case of OMAF-SRes-Partial, the user will see a blank area if the viewport moves away from the sent tiles. In the case of Viewport-Only, the user will start seeing a blank area if the viewport moves at all in relation to the requested one.

Because of this, it is necessary to develop a model in the context of the simulator that objectively evaluates the impact in video quality caused by a viewport mismatch, based on the work in [52].

The author [52] measured the degradation of the viewport quality when the rendered viewport does not match the requested viewport. In order to achieve the requested viewport center was fixed at (0° , 0°). Next, 36 viewport orientations uniformly distributed along the equator of the frame were selected for viewport extraction. For each video sequence and viewport orientation, the author rendered the first frame of each second of video. Each rendered viewport was then compared to the same viewport rendered from the original video by computing the V-PSNR for each frame and averaging the values for all frames of each sequence. For each viewport orientation, the resulting V-PSNR for MonoEqui was subtracted to the V-PSNR obtained for each delivery scheme, thus obtaining a V-PSNR gain relative to MonoEqui, ΔV-PSNR. Figure 4.9 shows the obtained values for the ΔV-PSNR for all streaming schemes with the longitude of the viewport center varying along the equator of the sphere.



(a) *ChairliftRide* (b) *SkateboardInLot*

(c) *KiteFlite*

Figure 4.9: ΔV-PSNR gain curves relative to MonoEqui for different delivery schemes [52].

4.7.2 Quality Impact Estimation

The data presented in Figure 4.9 can be used to estimate the quality impact when there is a viewport misalignment. The graphs are shown in Figure 4.9 in relation to MonoEqui. The reason why there are positive values is because the quality emphasis regions of the partial delivery schemes yield a higher V-PSNR value than MonoEqui for the same bitrate. In order for these values to be used to assess quality impact inside the same delivery scheme, they need to be normalized. This is done by, for each delivery scheme, subtracting the maximum ΔV-PSNR value of that scheme to each ΔV-PSNR value. This way, the maximum value is zero. This means that, at the corresponding angle, there is no quality impact on the video caused by a viewport mismatch.

The idea is that, in each TTI, for the segment that is currently being played, the angular difference between the actual viewport center and that segment's requested viewport center is calculated. Then, using the information from the graphs in Figure 4.9, the angular difference is used to estimate the corresponding quality impact. The quality impact, expressed as a gain in dB, is then used to calculate the adjusted V-PSNR. When calculating the adjusted V-PSNR, it is necessary to keep in mind that the quality impact will be made on top of the originally received video quality level. For this end, the information in Figure 4.6 is used in such a way that the gain from the quality impact is subtracted to the original V-PSNR value of the segment's video quality level, yielding the adjusted V-PSNR. Knowing that the V-PSNR values of the quality levels are equally spaced, it is then possible to map this adjusted

V-PSNR to the corresponding adjusted quality level. It was decided to use adjusted quality levels with a spacing of 0.1 in order to achieve a higher level of granularity when assessing the quality impact.

In order to illustrate this with an example, suppose that the client is streaming a segment from the *ChairliftRide* sequence at the maximum quality level using the OMAF-SRes scheme. In a certain TTI, there is an angular difference of 110° between the actual viewport center and that segment's requested viewport center. Looking at Figure 4.9, this corresponds to a ΔV-PSNR of -0.5 dB, which, when subtracted to the maximum ΔV-PSNR value for that video sequence and delivery scheme, gives a quality impact of -2.5 dB. Referencing Figure 4.6, this value is factored into the V-PSNR of the delivered video quality, giving an adjusted V-PSNR of approximately 37.3 dB. This gives an adjusted quality level of approximately 4.3.

In [52], the author did not include the ΔV-PSNR curves for the OMAF-SRes-Partial and Viewport-Only schemes. Therefore, these had to be estimated.

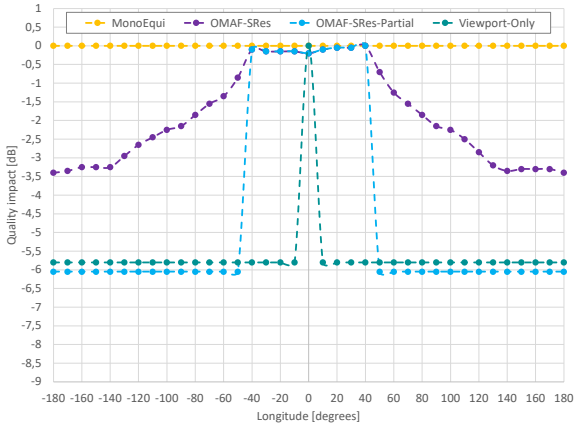
For the OMAF-SRes-Partial scheme, the quality impact is the same as for OMAF-SRes for angles where the viewport is completely covered with tiles. Considering the tile layout in Figure 3.9(a) and a 96° viewport horizontal FoV, it is concluded that the viewport is completely covered for angular differences between -42° and 42° . When the angular difference surpasses these values, the quality impact curve will be different because, instead of low resolution tiles, the user will start seeing a blank area, which has a much higher impact on perceived quality. Since formal research is lacking in this area, it was estimated that 10% is the maximum percentage of blank area of the viewport that the user is willing to tolerate. This corresponds to an angular difference lower or higher than approximately -52° or 52° , respectively. As such, beyond these values the quality impact should be such that, if segment is delivered with the maximum quality level, the adjusted quality level is the minimum one.

The estimated quality impact curve for the Viewport-Only scheme follows a similar logic. When there is no angular difference, there is no impact in quality. The difference is that, even at an angular difference of 1° there is already an impact. The estimated maximum blank area tolerable by the user was also estimated to be 10% of the viewport. At this percentage, the minimum and maximum angular difference tolerable by the user is approximately -10° and 10° , respectively. Figure 4.10 shows the estimated quality impact values for each delivery scheme. In the simulator, these values were interpolated in order to achieve a granularity of 1° in the angular difference.

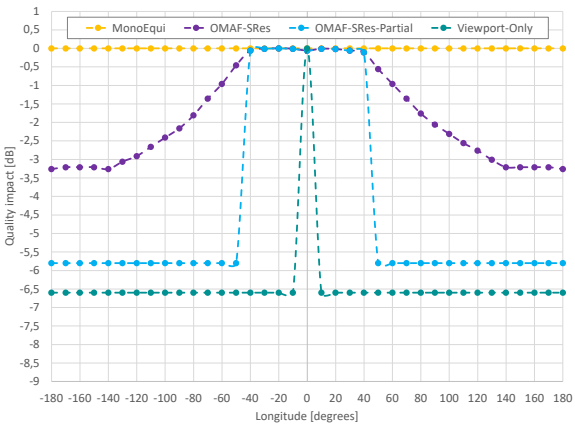
4.7.3 Viewport Trajectories

In order to accurately calculate the perceived video quality along a streaming session, it is also necessary to have information about the viewport trajectory along that streaming session. The head movement is one of the most important user behaviors, which can reflect their visual attention, preference and unique motion pattern. Therefore, it was decided to use a real head tracking dataset from [72].

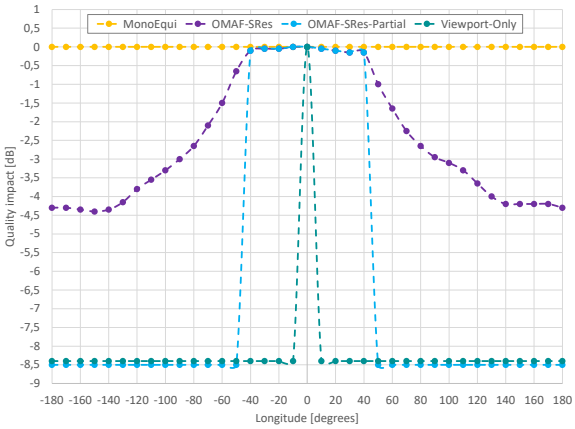
This dataset is composed of 48 users, each watching nine 360° videos on an HTC Vive headset. The participants are free to look around the video and their head motions are recorded during the viewing of these videos. For each participant and video, a list of timestamps, playback times, and quaternions



(a) *ChairliftRide*



(b) *SkateboardInLot*



(c) *KiteFlite*

Figure 4.10: Quality impact gain curves for different delivery schemes.

representing the viewing direction and coordinates of the HMD device in space is recorded.

Since the videos contemplated in this experiment are not the same as the ones used in the simulator, it was necessary to find the videos in the experiment that most closely resemble the ones described in Subsection 4.4.1. This was done by evaluating the amount of motion content of each video. Taking this into account, it was decided to match the *ChairliftRide* sequence with the trajectories from the *The Last of the Rhinos* sequence, the *SkateboardInLot* sequence with the trajectories from the *Freestyle Skiing* sequence and the *KiteFlite* sequence with the trajectories from the *360° Cooking Battle* sequence.

In the simulator, only the longitude is considered. This was done not only because of computational complexity reasons, but also because during a 360° video viewing session, the user tends to concentrate their attention in areas around the equator the vast majority of the time, meaning that the latitude is almost always zero or close to that value. Figure 4.11 shows the density maps of viewers' viewing directions for three different videos, which confirms this notion.

Since the dataset is recorded using unit quaternions (qx, qy, qz, qw) representing rotations in 3D space, these values had to be converted to a time series of longitudes in order to more easily calculate the angular differences due to viewport mismatch. The used system adopts a left-handed coordinate system

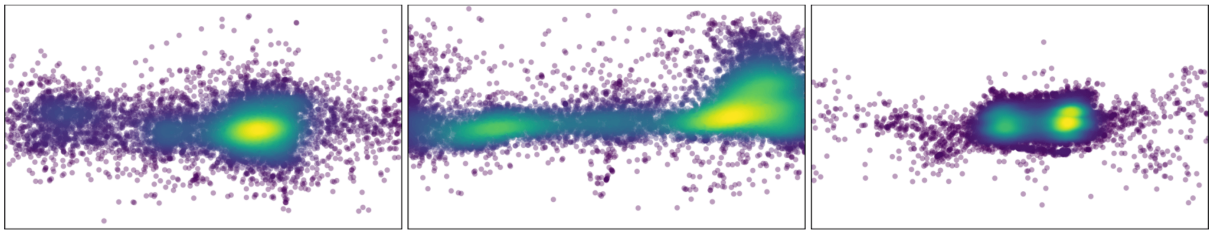


Figure 4.11: Density maps of participants' gazing directions [72].

in which the positive x, y and z axes point right, up and forward, respectively. The unit vector representing which direction a user is looking at can be calculated from

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \cdot qx \cdot qz + 2 \cdot qy \cdot qw \\ 2 \cdot qy \cdot qz - 2 \cdot qx \cdot qw \\ 1 - 2 \cdot qx^2 - 2 \cdot qy^2 \end{bmatrix}. \tag{4.4}$$

The unit vector can then be used to calculate the longitude with a transformation from cartesian to spherical coordinates where

$$\lambda = \arctan \frac{z}{x}, \tag{4.5}$$

where λ is the longitude in degrees.

Figure 4.12 shows the longitude evolution along one of the videos for one of the participants. It is possible to observe that this particular participant spends a significant period of time looking in the same direction. However, there are also short bursts of time where the participant moves their head very quickly. There is one instance where the participant looks all the way around the sphere in a period no longer than 5 seconds. These are the situations that are most likely to challenge the partial delivery schemes in terms of keeping the requested viewport within a reasonable distance of the actual viewport.

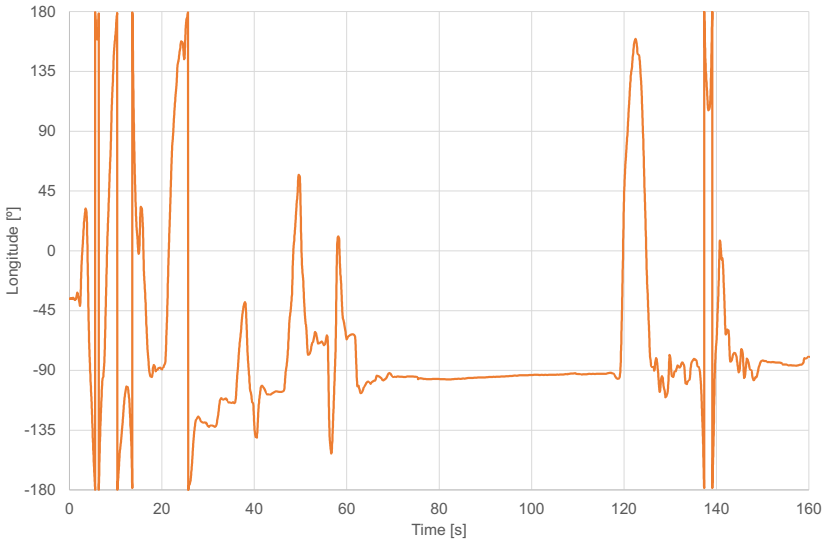


Figure 4.12: Longitude evolution for a user along a viewing session.

4.8 QoE Model

In order to assess whether the users are satisfied or not with the streaming session, a model that estimates the QoE of each user must be used. It was decided to use a parametric QoE model that depends on the following metrics: duration of the initial delay, frequency and duration of rebuffering events, and average and standard deviation of the served video quality. This model is based on the work presented in [73]. Although this model was designed to calculate the QoE for 2D video, it was used because there is a lack of formal research into parametric QoE models for 360° videos.

The QoE for user i is calculated as

$$\mathrm{QOE}_i = 5.67 \times \frac{\overline{q_i}}{q_{max}} - 6.72 \times \frac{\hat{q}_i}{q_{max}} + 0.17 - 4.95 \times F_i, \tag{4.6}$$

where $\overline{q_i}$ is the average served quality level, $\hat{q}_i$ is the standard deviation of the served quality level, both normalized to the number of available quality levels in the server, q_{max}, and F_i models the influence of the rebuffering events and is calculated as

$$F_i = \frac{7}{8} \times \max\left(\frac{\ln(\phi_i)}{6} + 1, 0\right) + \frac{1}{8} \times \frac{\min(\psi_i, 15)}{15}, \tag{4.7}$$

where ϕ_i is the freeze frequency and ψ_i is the average freeze duration. ϕ_i is calculated by dividing the number of rebuffering events by the total duration of the streaming session and ψ_i is calculated by dividing the sum of the durations of each rebuffering event by the total duration of the streaming session.

The $\overline{q_i}$ and $\hat{q}_i$ values can be calculated according to

$$\overline{q_i} = \frac{\sum_{k=1}^{K} Q_{L_k}}{K}, \tag{4.8a}$$

$$\hat{q}_i = \sqrt{\frac{\sum_{k=1}^{K} (Q_{L_k} - \overline{q_i})}{K}}, \tag{4.8b}$$

where K is the total number of downloaded segments and Q_{L_k} is the quality level of the k^{th} segment.

The coefficients in (4.6) and (4.7) have been fixed according to the work in [74] and [75]. In [74], the authors made a subjective test with 7 video quality levels in order to fix the coefficients. This is the reason why it was decided to also use 7 quality levels in the simulator.

A few adaptations were made to the presented model in order to make it more suitable to calculate the QoE for 360° video.

First, the presented QoE model does not take the initial buffering delay into account. However, this metric is quite impactful on the QoE, as described in Section 3.7.2. In order to incorporate this parameter into the model, it is included in the calculation of F_i. It was decided that it should not count toward the frequency of rebuffering events, as it happens at the beginning of the session and does not interrupt the

video playback, not being as annoying to the user. Therefore, it only counts toward the calculation of the mean rebuffering duration.

Second, the presented model does not take the quality impact due to the viewport mismatch described in Section 4.7 into account. Therefore, it was decided that the $\overline{q_i}$ and $\hat{q}_i$ should be calculated not based on the served video quality levels, but on the adjusted quality levels calculated from taking the quality impact of the viewport mismatch into account. Furthermore, it was decided that, for the OMAF-SRes-Partial and Viewport-Only schemes, a blank area in the viewport should impact the QoE at the level of the stalling event. Therefore, whenever the blank area surpasses the percentages of the viewport area discussed in Subsection 4.7.2, it counts towards the freeze frequency, ϕ_i, and average freeze duration, ψ_i. Naturally, this is only an adaptation, as the original QoE model was not developed for 360° video. Therefore, the absolute QoE value may not match the results of a subjective test. However, this adaptation achieves the desired effect of showing the relative differences in QoE between different delivery schemes and system conditions.

With this model, it is possible to analyze the QoE of each user in each TTI of the streaming session, which allows the study of the evolution of the achieved QoE up to a certain moment in the session, and the global QoE, taking into account the entire duration of that session. By looking at the global QoEs of different streaming schemes, it is possible to evaluate and compare the corresponding users' satisfaction. Analyzing (4.6) and (4.7) it is possible to observe that the theoretical range of the QoE is [0.00, 5.84], although in practice it is observed that it does not surpass 5. This closely corresponds to the typical levels of a MOS. Table 4.8 relates the QoE with the perceived quality.

Table 4.8: Perceived quality by user i with a given global QoE.

QoE	Perceived quality
$4 \leq \mathrm{QoE}_i \leq 5$	Excellent
$3 \leq \mathrm{QoE}_i < 4$	Good
$2 \leq \mathrm{QoE}_i < 3$	Fair
$1 \leq \mathrm{QoE}_i < 2$	Poor
$0 \leq \mathrm{QoE}_i < 1$	Bad

4.9 Simulator Flowchart

The complete flowchart of the simulator, including the all the components described in the sections above, is presented in Figure 4.13.

First, the simulation settings are imported. The main settings include the total simulation time, which is equivalent to the duration of the streaming session to be simulated, the number of connected users, the number of PRBs to be allocated per TTI, the end-to-end latency, the number of segments to request when initially buffering or rebuffering, and which delivery scenario is being simulated. The simulator then imports the bitrates associated to the quality levels of the selected delivery scenario. The simulator also imports the dataset of the users' viewport trajectories, and the evolution of their CQIs over time.

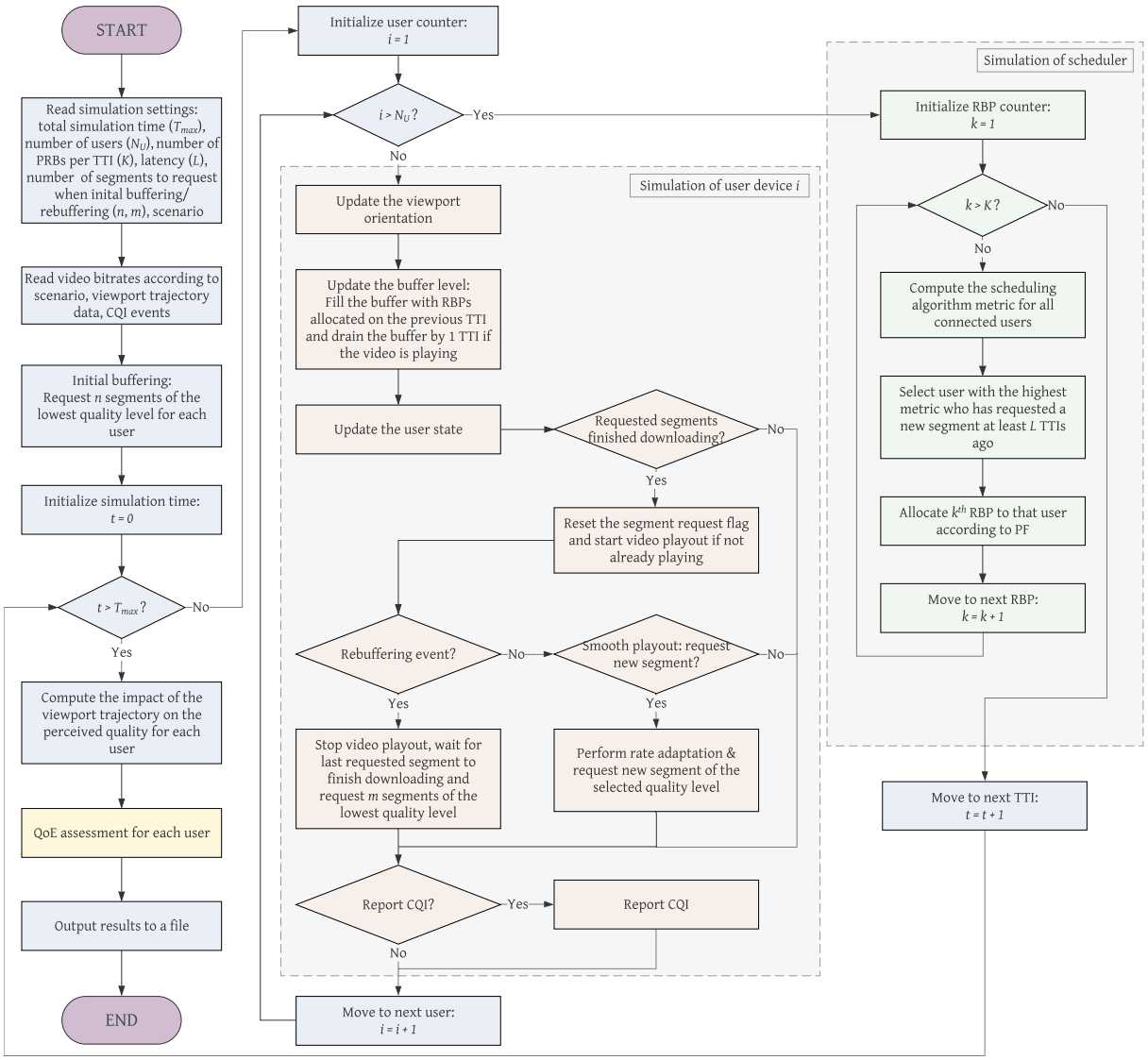


Figure 4.13: Complete simulator flowchart.

After all the data are imported, the simulator sets all users to request their initial segments, which will be of the lowest quality available. The simulator then enters its main *for* loop. The length of this loop is equivalent to the number of TTIs one wishes to simulate, which is given by the total simulation time and the duration of each TTI. In the case of the developed simulator, it is assumed that each TTI is 1 ms in length. For each iteration, the state of each user's device is simulated: the viewport orientation is updated, the buffer level is updated, the rate adaptation algorithm may request new segments to the server and choose their quality according to the QAAD, and the CQI may be reported to the base station.

Following the simulation of all users' devices, the resource scheduler is simulated. In each TTI, the PF scheduling algorithm distributes the available PRBs among the users that are requesting data.

When the simulator reaches the predefined simulation time, the impact of the viewport trajectory on the perceived video quality by each user is computed, and this and other data are used as input to the model that assesses the QoE for each user.

Chapter 5

Simulations and Results

This chapter presents the employed methodology to compare the performances of the various delivery schemes and the main results from the simulations. In Section 5.1, the simulation parameters are described. In Section 5.2, the outputs of the simulation are presented. In Section 5.3, the sensitivity analysis is presented. In Section 5.4, the performances of the streaming schemes are compared. Finally, in Section 5.5, a summary of the main results is presented.

5.1 Simulation Parameters

In order to study how the different streaming schemes described in Section 4.4 compare against each other, it is necessary to run the developed simulator several times, changing the used scheme each time. However, in order to make an accurate comparison, all the other simulation parameters need to be the same, so that any changes in the outcome are due only to the used streaming scheme.

For this end, each simulation consists in a streaming session with a duration of 3 minutes. During this period, a fixed number of users are simultaneously streaming 360° video using one of the aforementioned streaming schemes with a fixed end-to-end latency, which may take one of the values in Table 4.7. Each user is streaming one of the three available test sequences described in Subsection 4.4.1. The bitrates for each quality level available in the server are presented in Tables 4.4, 4.5 and 4.6. The allocation of video to each user is made randomly at the beginning of the simulation using a uniform distribution.

The test sequences are available in the server in 1 second segments, except in the case of the Viewport-Only scheme. Since the server sends the video to the client frame by frame, it is considered that each segment has 40 ms, which is the duration of a frame in a 25 frames/s video.

The Monolithic scheme requests 5 segments initially and 5 segments after a stall, all with the minimum quality available. The other schemes request 1 segment initially and 1 segment after a stall, also with the minimum quality available. This is due to the user's head motion influencing the video quality, as it would worsen the user's experience to watch consecutive segments with the same viewing direction.

Each user starts requesting their initial segments at a random time between 0 and 200 ms. This step was taken in order to prevent a situation in which all users would start requesting segments at the exact same time, which would unnecessarily overload the network. Furthermore, having users start their streaming sessions at different times better represents a real scenario.

After receiving the initial segments with minimum quality, the DASH client selects the quality of the subsequent segments using the QAAD algorithm, described in Section 4.3. As described in this section, this algorithm has two parameters, μ and σ, that need to be set. The marginal buffer length, μ, defines the minimum length the buffer needs to have in order for the requested quality level to be increased. The minimal buffer length, σ, defines the critical buffer length that the client always tries to keep, triggering a quality decrease if that condition is not met. Considering that the buffer threshold that triggers the request of the next segment, α, described in Section 4.3, influences the range of values the buffer can have, it makes sense to set μ and σ dynamically. Therefore, μ is set to 80% of α and σ is set to 20% of α. Considering a buffer threshold, α, of 800 ms, μ would be 640 ms and σ would be 160 ms.

Each user is allocated one of the CQI profiles from the dataset described in Section 4.2. This allocation is made as a random permutation, meaning that each user is randomly assigned one CQI profile without repetitions. The viewport trajectory dataset described in Section 4.7 is allocated to each user in the same way. Considering that there are 3 test sequences, 200 CQI profiles and 48 viewport trajectories available, this equates to 28800 different combinations.

It was decided that the simulator would use a network configuration with 2x2 MIMO, 15 KHz SCS, and 20 MHz bandwidth. According to Table 2.3, this means that there are 106 PRBs available per TTI, and according to Table 2.2, each TTI is 1 ms in length. The TTI is the basic time unit of the simulator. As such, with a 3 minute simulation, 180000 TTIs are simulated. The throughput for each CQI can be seen in Table 4.1. In ideal conditions, the throughput is 170 Mbps which is low in comparison to what 5G is able to achieve. This option was taken in order to test the various scenarios in conditions similar to what early deployments of 5G networks may present.

As described, some of the simulator parameters, such as the test sequence, CQI profile and viewport trajectory allocations, are stochastic processes. Naturally, this results in a random component of the input variables, which introduces a random component in the simulation results. Therefore, it is not sufficient to run the simulation once, as the accuracy of that result would be unknown. In order to obtain statistically relevant results, the Monte Carlo method is used. Each simulation is run a certain number of times with different random test sequence, CQI profile and viewport trajectory allocations each time, in order to derive a final result that includes a mean value and a confidence interval. The Monte Carlo method is described in Annex D.

5.2 Simulation Outputs

In order to assess the number of satisfied and non-satisfied users, the main output of the simulation is the final QoE of each user. The term ”final QoE” refers to the overall QoE level a user has at the end of the streaming session.

As described in Section 4.8, the used QoE model has two main factors contributing to the final value: metrics related to the radio conditions of the channel, such as average served quality, stall frequency, among others, and metrics related to the viewport mismatch due to the user’s head movement, such as

the average adjusted quality. These two factors are independent, meaning that, for example, the user can have a poor final QoE because the video was not served correctly due to the channel conditions, or because the video was correctly served but the head movement caused a high level of viewport mismatch throughout the streaming session. Since it is important to evaluate the impact that each of the main factors has on the final QoE, this metric is computed in two steps: first, only the metrics related to the radio conditions are taken into account, producing the QoE_{radio} metric. Then, the metrics related to the viewport mismatch are taken into account on top of the radio metrics, producing the QoE_{final} metric. This way, it is possible to assess to which extent each of the two factors contribute to the final QoE. Figure 5.1 illustrates how these metrics are calculated.

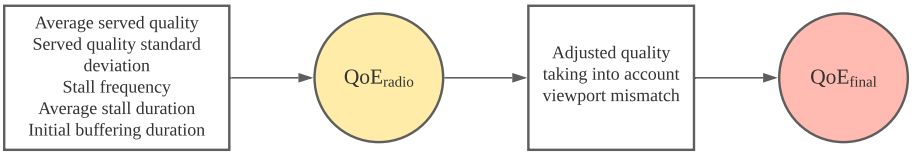


Figure 5.1: QoE metrics calculation.

Performance is assessed in terms of the number of users that are satisfied and/or non-satisfied at the end of a streaming session, out of all the connected users. It is considered that a user is satisfied if their QoE_{final} is above a certain limit S ($QoE_{final} \geq S$) and non-satisfied if their QoE_{final} is below a certain limit NS ($QoE_{final} \leq NS$), where $S > NS$. For a given number of connected users, the percentage of satisfied users for that S is denoted by $\%_S$ and the percentage of non-satisfied users for that NS is denoted by $\%_{NS}$. Naturally, these values depend heavily on the chosen simulation parameters. However, if the parameters are kept constant, the relative performance can be used to make assessments.

5.3 Sensitivity Analysis

As explained in Section 4.3, it is necessary to perform a sensitivity analysis to the buffer threshold that triggers the request of the next segment, α. Because this value is different for different end-to-end latency values, L, it is also necessary to perform the analysis taking this parameter into account.

In order to do this, a series of values of α are simulated for each of the values of latency, L, 10 ms and 1 ms, which are equivalent to having MEC infrastructure deployed or not, as explained in Section 4.6. This simulation is performed for various numbers of connected users, in order to assess the performance limits of each streaming scenario.

5.3.1 MonoEqui Scheme

L = 10 ms (without MEC)

Figures E.1 and E.2 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for the MonoEqui scheme, with a latency of 10 ms. Only the value

of α = 6000 ms is simulated because the MonoEqui is a viewport-independent scheme, and therefore the best solution is to fill the buffer as much as possible. It is possible to observe that there are no differences between the QoE_{radio} and the QoE_{final}. This is also because, since MonoEqui is a viewport-independent scheme, the video quality is uniformly distributed throughout the 360° sphere, and therefore the viewport trajectory does not impact the user's QoE.

For each plotted point of the curve in Figure E.1, at least n Monte Carlo simulations were performed to guarantee with 95% certainty that the real solution does not differ by more than 1% from the real values. The percentages of satisfied users were computed by averaging the percentages obtained in each simulation. In Figure E.2, it was not possible to perform the minimum number of simulations because the computational costs were too high to perform the simulations in a reasonable period. However, a sufficiently high number of simulations were performed in order to be able to estimate the general position of the curve.

As expected, the percentage of satisfied users decreases as the number of connected users increases. Conversely, the percentage of non-satisfied users increases. Table 5.1 presents the capacity of this configuration, where the second column shows the maximum number of connected users for which 90% have a $QoE_{final} \geq 3$ and the third column shows the maximum number of connected users for which 5% have a $QoE_{final} \leq 2$. The total capacity is deduced from the metric that has the least capacity.

Table 5.1: Capacity of MonoEqui for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 10 ms.

α **[ms]**	**Maximum number of connected users for which S = 3; $\%_S$ = 90%**	**Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%**	**Capacity**
6000	**48.57**	63.33	48.57

L = 1 ms (with MEC)

Figures E.3 and E.4 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for the MonoEqui scheme, with a latency of 1 ms.

The results are fairly similar to the results without MEC. As it is possible to observe in Table 5.2, only the number of maximum number of connected users for which 5% have a $QoE_{final} \leq 2$ increases slightly. The reason for the identical results is that the segment length of 1 second is orders of magnitude larger than the tested latency values and the tested α value is also much larger. Such a small change to the latency in relation to these values almost makes no difference in the capacity of the scheme.

Table 5.2: Capacity of MonoEqui for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 1 ms.

α **[ms]**	**Maximum number of connected users for which S = 3; $\%_S$ = 90%**	**Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%**	**Capacity**
6000	**48.57**	64	48.57

5.3.2 OMAF-SRes Scheme

L = 10 ms (without MEC)

Figures E.5 and E.6 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the OMAF-SRes scheme, with a latency of 10 ms. As it is possible to observe in Figure E.5(a), the number of users with a good QoE_{radio} increases with the increase of α, which is expected. With the highest level tested, α = 6000 ms, which is the same as in MonoEqui, the results are better with this scheme because the bitrates of the test sequences are lower and therefore more users can be served. However, when observing Figure E.5(b), it is possible to see that this value of α is not the best, since such a high buffer threshold causes the high quality tiles to be the wrong ones in terms of where the viewport is when the users views the segment. With lower α values, it is possible to see that the penalization between the QoE_{radio} and the QoE_{final} is lower.

Table 5.3 shows the capacity of this configuration. Although the capacity is the same for both α = 900 ms and α = 1000 ms, the second value is the best one since it supports a higher number of connected users for which 90% have a $QoE_{final} \geq 3$.

Table 5.3: Capacity of OMAF-SRes for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 10 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
50	3.09	**2.8**	2.8
100	**10.63**	11.18	10.63
300	**22.13**	30.34	22.13
500	**34.17**	40.33	34.17
700	**42.33**	43.57	42.33
900	46.67	**46.25**	46.25
1000	50.11	**46.25**	46.25
6000	**20**	70.91	20

L = 1 ms (with MEC)

Figures E.7 and E.8 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the OMAF-SRes scheme, with a latency of 1 ms.

The results are fairly similar to the results without MEC. Similarly to MonoEqui, this happens because the segment length and the buffer threshold are much larger than the latencies. As it is possible to observe in Table 5.4, the capacities are, in general, slightly better. Furthermore, it is possible to observe that the biggest improvement is for α = 50 ms, where the capacity is 16% higher. For α = 6000 ms, no users are supported because at most only 89% of the total users are satisfied, which is lower than the desired $\%_S$ = 90%. This happens because the QoE_{final} improvement due to the lower viewport mismatch is not enough to offset the lower average QoE_{radio} that was observed.

Table 5.4: Capacity of OMAF-SRes for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 1 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
50	**3.25**	4	3.25
100	11.96	**11.72**	11.72
300	**20.91**	30	20.91
500	**38.33**	40.35	38.33
700	**42.38**	43.85	42.38
900	46.67	**46.25**	46.25
1000	50.11	**47.14**	47.14
6000	<**1**	65	<1

5.3.3 OMAF-SRes-Partial Scheme

L = 10 ms (without MEC)

Figures E.9 and E.10 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the OMAF-SRes-Partial scheme, with a latency of 10 ms. As it is possible to observe in Figure E.9(a), the percentage of satisfied users in terms of QoE_{radio} is larger than with OMAF-SRes, due to the lower bitrates. However, the OMAF-SRes-Partial is much stricter in terms of the QoE_{final} penalization because of the viewport mismatch. Since only some tiles are delivered with this scheme, the QoE model considers a stall when the viewport mismatch is higher than $\pm 45^{\circ}$, as explained in Section 4.8, which heavily impacts this metric. The delivered viewing direction is constant along each segment, while the viewport orientation of the user is continually changing, leading to a high number of these events. Therefore, as it is possible to observe in Figure E.9(b), the impact on the QoE_{final} is very high and practically no users are satisfied.

Table 5.5 shows the capacity of this configuration. As it possible to observe, no users are supported with these requirements.

Table 5.5: Capacity of OMAF-SRes-Partial for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 10 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
50	<1	<1	<1
100	<1	<1	<1
300	<1	<1	<1
500	<1	<1	<1
700	<1	<1	<1
900	<1	<1	<1
1000	<1	<1	<1
6000	<1	<1	<1

L = 1 ms (with MEC)

Figures E.11 and E.12 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the OMAF-SRes-Partial scheme, with a latency of 1 ms.

The results are fairly similar to the results without MEC. The reduction in latency is not enough to offset the QoE_{final} penalization, as the problem of the segment length and buffer threshold being much larger than the latency persists. The ideal situation would be to request the next segment as close as possible to the current one ending. This way, the viewing direction would be as updated as possible. However, the problem of keeping the viewing direction for the entire duration of the 1 second segment persists, which is detrimental to the QoE_{final}. Also, the bitrates are still too high to allow for a low buffer threshold. As it is possible to observe in Table 5.6, still no users are supported.

Table 5.6: Capacity of OMAF-SRes-Partial for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 1 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
50	<1	<1	<1
100	<1	<1	<1
300	<1	<1	<1
500	<1	<1	<1
700	<1	<1	<1
900	<1	<1	<1
1000	<1	<1	<1
6000	<1	<1	<1

5.3.4 Viewport-Only Scheme

L = 10 ms (without MEC)

Figures E.13 and E.14 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the Viewport-Only scheme, with a latency of 10 ms. Since the Viewport-Only scheme uses a segment length of 40 ms, the tested α values are much smaller in order to reflect this change. In Figure E.13(a), it is possible to observe that no users are satisfied for $\alpha \leq$ 10 ms. This is expected, since it means the client has 10 ms to receive the next segment, and the latency alone is 10 ms. Since the involved bitrates are the smallest of all the schemes, a value of $\alpha \geq$ 15 ms is enough to support 100% of satisfied users in terms of QoE_{radio} for more than the maximum tested number of connected users. As it is possible to observe in Figure E.13(b), similarly to the OMAF-SRes-Partial scheme, there is also a high penalty in the QoE_{final}, with the lower values of α being better, as expected. However, the penalty is not as high as with OMAF-SRes-Partial. This is because, contrarily to what happens with this scheme, where the viewing direction is constant throughout each 1 second segment, in the case of Viewport-Only the viewing direction is updated every frame, since each segment has a length of 40 ms. This way, even though the QoE model considers a

stall with a viewport mismatch that is more than $\pm 10^{\circ}$, it is possible to obtain better results.

Table 5.7 shows the capacity of this configuration. As it possible to observe, none of the α values fulfills the minimum requirements in terms of satisfied and non-satisfied users. However, the value α = 15 ms is the one that presents the highest percentages of satisfied users and the lowest percentages of non-satisfied users.

Table 5.7: Capacity of Viewport-Only for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 10 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
1	<10	<10	<10
2	<10	<10	<10
3	<10	<10	<10
4	<10	<10	<10
5	<10	<10	<10
10	<10	<10	<10
15	<10	<10	<10
20	<10	<10	<10
30	<10	<10	<10
40	<10	<10	<10

L = 1 ms (with MEC)

Figures E.15 and E.16 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α. Similarly to the L = 10 ms case, it is possible to observe in Figure E.15(a) the value of α = 1 ms is too low to have any satisfied users in terms of QoE_{radio} since the latency is also 1 ms. With α = 2 ms, it already possible to support some users. For $\alpha \geq$ 3 ms, it is possible to support 100% of satisfied users in terms of QoE_{radio} for more than the maximum tested number of connected users. In Figure E.15(b), it is possible to observe that with the Viewport-Only scheme, changing the latency from 10 ms to 1 ms makes a big difference, as the segment length and the buffer thresholds are in the same order of magnitude. For the best case of α = 3 ms, there is only a 3 ms delay between the requested viewing direction and the actual viewing direction. Even with very strict requirements in terms of viewport mismatch, this is a low enough delay in order to have a low enough penalization in the QoE_{final}. As expected, the penalization starts to increase as α increases.

Table 5.8 shows the capacity for this configuration. Although for α = 2 ms and α = 3 ms it is possible to have a number of connected users for which 5% have a $QoE_{final} \leq$ 2 that is higher than the minimum number of users tested, for S = 3, $\%_{NS}$ = 90%, this is not true. Therefore, the capacity of this scheme does not surpass the minimum number of users tested, although by observing Figure E.15(b) it is possible to see that for α = 3 ms it is almost possible to reach 90% of satisfied users for even the maximum number of users tested, which is a very promising result when comparing to the other schemes.

Table 5.8: Capacity of Viewport-Only for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 1 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
1	<10	<10	<10
2	**<10**	41.25	<10
3	**<10**	>100	<10
4	<10	<10	<10
5	<10	<10	<10
10	<10	<10	<10
15	<10	<10	<10
20	<10	<10	<10
30	<10	<10	<10
40	<10	<10	<10

5.3.5 Viewport-Only-Margin

The Viewport-Only-Margin scheme aims at fixing the shortcomings of the Viewport-Only scheme, by incorporating some of the ideas from the OMAF-SRes-Partial scheme and the work described in [50]. The ideia is to modify the Viewport-Only scheme in order to include a margin around the sent viewport. This way, if the user moves their head inside one segment, there is an area of tolerance before the viewport starts to lose coverage, avoiding a stall in the QoE model, which improves the QoE_{final}. This is comparable to the OMAF-SRes-Partial scheme, but in this case the segment length is 40 ms instead of 1 s. Naturally, adding a margin to the sent viewport increases the video bitrate which worsens the QoE_{radio}, so there is trade-off. However, with Viewport-Only, the involved bitrates are low enough so that 100% of satisfied users are supported for more than the maximum number tested of connected users.

The test sequences have a spatial resolution of 7680×3840 pixels, while the viewport has a resolution of 2000×2000 pixels. By adding a 107 pixel margin around the sent viewport, which equates to an extra 5° FoV in every direction, the sent video will have a resolution of 2214×2214 pixels. This new resolution has an area that is 16.6% of the original test sequence. By following the steps described in Subsection 4.4.2, the estimated bitrates for each quality level and test sequence are the ones presented in Table 5.9.

Table 5.9: Estimated bitrates for each quality level and test sequence. Scheme: Viewport-Only-Margin.

Quality level	**1**	**2**	**3**	**4**	**5**	**6**	**7**
ChairliftRide [Mbps]	0.163	0.227	0.312	0.427	0.581	0.784	1.051
SkateboardInLot [Mbps]	0.160	0.223	0.309	0.424	0.557	0.779	1.042
KiteFlite [Mbps]	0.159	0.224	0.311	0.426	0.578	0.775	1.030

The quality impact gain curves described in Subsection 4.7.2 also need to be adapted in order to reflect the extra margin around the viewport. With the Viewport-Only scheme there is a ±10° tolerance before

the QoE model considers a stall. With an extra 5° per side, this limit becomes ±15° . Between -10° and 10° of viewport mismatch, the quality impact is the same as with MonoEqui.

L = 10 ms (without MEC)

Figures E.17 and E.18 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the Viewport-Only-Margin scheme, with a latency of 10 ms. For this simulation, the values α = 15 ms and α = 20 ms were chosen, as these were the two best values in the Viewport-Only scheme. As it is possible to observe in Figure E.17(a), the higher associated bitrates due to the margins around the viewport still allow for 100% of users with a $QoE_{radio} \geq 3$. In Figure E.17(b), it is possible to see that the 5° of extra margin are enough to make percentage of users with $QoE_{final} \geq 3$ go from around 50% in Viewport-Only, in Figure E.13(b), to around 90% with this configuration. Also, the percentage of users with $QoE_{final} \leq 2$ goes from between 20% and 30% to around 5%.

Table 5.10 shows the capacity of this configuration. The improvements in the users' QoE values are enough to allow the configuration with α = 15 ms to have a capacity of more than the maximum number of users tested, whereas with the Viewport-Only scheme, it was not possible to support any users.

Table 5.10: Capacity of Viewport-Only-Margin for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 10 ms.

α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
15	>100	>100	>100
20	**<10**	>100	<10

L = 1 ms (with MEC)

Figures E.19 and E.20 plot the percentage of satisfied and non-satisfied users, respectively, as a function of the number of connected users, for various values of α, for the Viewport-Only-Margin scheme, with a latency of 1 ms. For this simulation, the values α = 3 ms, α = 4 ms and α = 5 ms were chosen, as these were the three best values in the Viewport-Only scheme. As it is possible to observe in Figure E.19(a), the higher associated bitrates due to the margins around the viewport start to degrade the QoE_{radio} when 100 users are connected with α = 3 ms. In Figure E.19(b) it is possible to observe that the extra 5° of margin are enough to have almost 100% of satisfied users and no non-satisfied users, as it possible to see in Figure E.20.

Table 5.11 shows the capacity of this configuration. The improvements in the users' QoE values are enough to allow the configurations with all values of α to have a capacity of more than the maximum number of users tested, whereas with the Viewport-Only scheme, it was not possible to support any users with these requirements. The best value is α = 4 ms because it is the one that supports a higher percentage of satisfied users. The best value of Viewport-Only is α = 3 ms, but because of the higher bitrates, this is not the case anymore.

Table 5.11: Capacity of Viewport-Only-Margin for S = 3, $\%_S$ = 90%, NS = 2, $\%_{NS}$ = 5%. L = 1 ms.

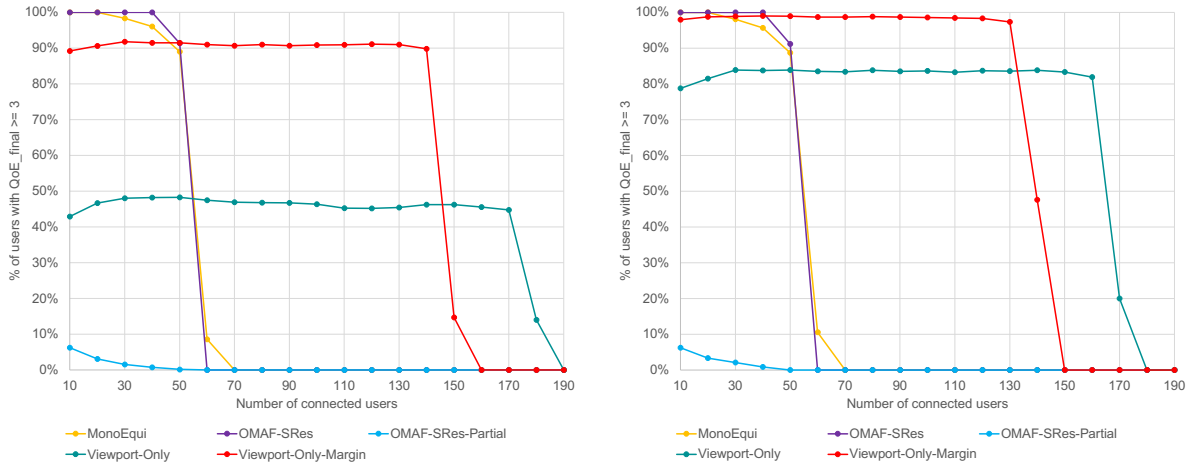
α [ms]	Maximum number of connected users for which S = 3; $\%_S$ = 90%	Maximum number of connected users for which NS = 2; $\%_{NS}$ = 5%	Capacity
3	>100	>100	>100
4	>100	>100	>100
5	>100	>100	>100

5.4 Schemes Comparison

In this section, the different schemes are compared against each other regarding the number of satisfied and non-satisfied users. In order to do this, the best α value for each scheme, estimated in Section 5.3, is used. In order to more accurately compare the Viewport-Only and Viewport-Only-Margin schemes, the simulations were re-run with up to 190 users, with 10 user increments.

5.4.1 Comparison of the Number of Satisfied Users (S = 3)

Figures 5.2(a) and 5.2(b) plot the percentage of satisfied users (S = 3), for all schemes, with a latency of 10 ms and 1 ms, respectively.



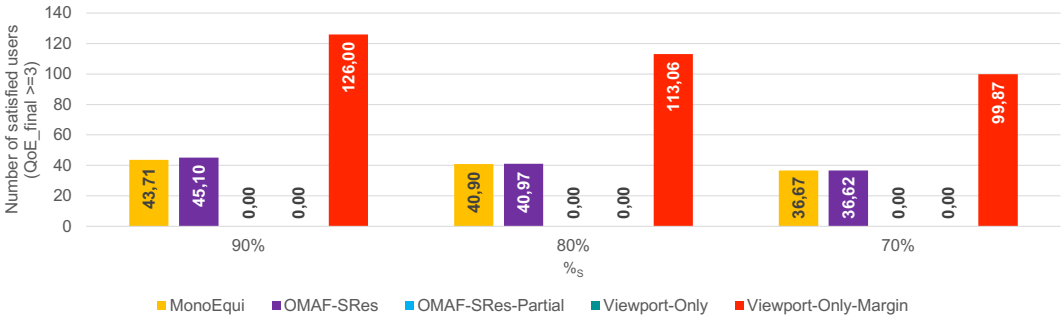
(a) L = 10 ms

(b) L = 1 ms

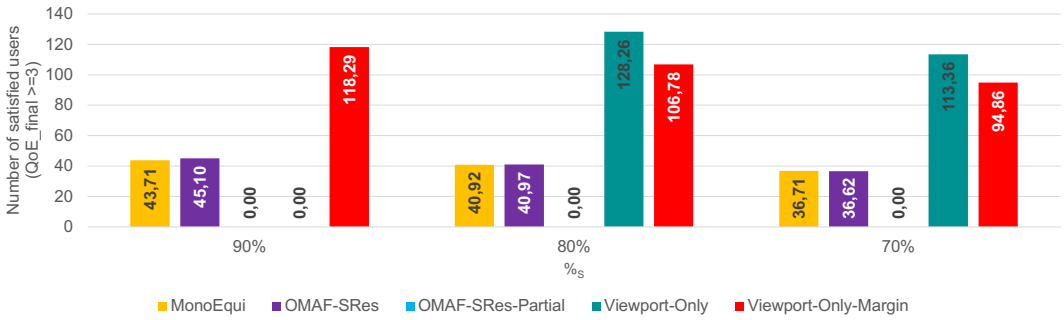
Figure 5.2: Percentage of satisfied users (S = 3) for all schemes.

Figures 5.3(a) and 5.3(b) present the number of satisfied users (S = 3), for all schemes, with a latency of 10 ms and 1 ms, respectively. For various percentages of satisfied users ($\%_S$ = 90%, 80%, 70%), the schemes support different numbers of satisfied users. These values are obtained by multiplying the percentage by the number of connected users at that percentage.

According to Figure 5.3, the scheme that supports the most users with a latency of 10 ms is Viewport-Only-Margin. For a latency of 1 ms, Viewport-Only supports more users than Viewport-Only-Margin for a percentage of satisfied users of 80% and 70%. The Viewport-Only scheme only has a number of satisfied users higher than zero for a latency of 1 ms and percentage of satisfied users of 80% and



(a) L = 10 ms



(b) L = 1 ms

Figure 5.3: Number of satisfied users (S = 3) for all schemes, $\%_S$ = 90%, 80% and 70%.

70%. It is also possible to observe that the OMAF-SRes scheme generally has a higher number of satisfied users than MonoEqui, except when the percentage of satisfied users is 70%, for both values of latency. The OMAF-SRes-Partial has zero satisfied users for any percentage of satisfied users and latency values.

Tables 5.12 and 5.13 present for each scheme the number of satisfied users (S = 3), the total number of connected users and the capacity gains relative to MonoEqui, for S = 3 and $\%_S$ = 90%, 80% and 70%, for a latency of 10 ms and 1 ms, respectively.

Table 5.12: Capacity of all schemes for S = 3, $\%_S$ = 90%, 80% and 70%. L = 10 ms.

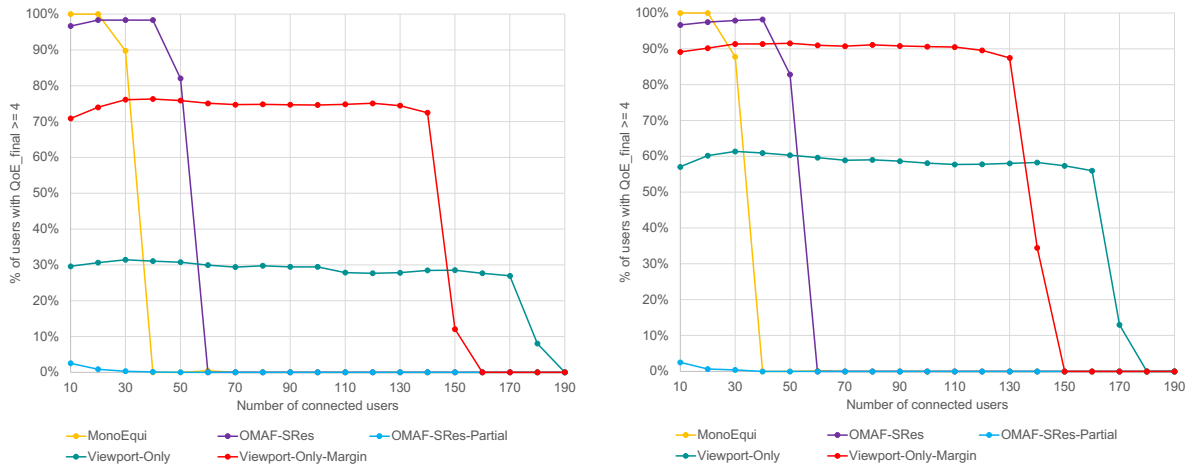
$\%_S$	Scheme (Satisfied users/Connected users/Gain)				
	MonoEqui	**OMAF-SRes**	**OMAFSRes-Partial**	**Viewport-Only**	**Viewport-Only-Margin**
90%	**43.71** 48.57 (0.00%)	**45.10** 50.11 (+3.18%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**126.00** 140.00 (+188.26%)
80%	**40.90** 51.13 (0.00%)	**40.97** 51.21 (+0.17%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**113.06** 141.33 (+176.43%)
70%	**36.67** 52.38 (0.00%)	**36.62** 52.31 (-0.14%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**99.87** 142.67 (+172.35%)

Table 5.13: Capacity of all schemes for S = 3, $\%_S$ = 90%, 80% and 70%. L = 1 ms.

$\%_S$	Scheme (Satisfied users/Connected users/Gain)				
	MonoEqui	**OMAF-SRes**	**OMAFSRes-Partial**	**Viewport-Only**	**Viewport-Only-Margin**
90%	**43.71** 48.57 (0.00%)	**45.10** 50.11 (+3.18%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**118.29** 131.43 (+170.62%)
80%	**40.92** 51.15 (0.00%)	**40.97** 51.21 (+0.12%)	**0.00** 0.00 (-100.00%)	**128.26** 160.32 (+213.44%)	**106.78** 133.47 (+160.94%)
70%	**36.71** 52.44 (0.00%)	**36.62** 52.31 (-0.25%)	**0.00** 0.00 (-100.00%)	**113.26** 161.94 (+208.25%)	**94.86** 135.51 (+158.40%)

5.4.2 Comparison of the Number of Very Satisfied Users (S = 4)

As described in Table 4.8, a QoE between 3 and 4 corresponds to a perceived quality of "Good", while a QoE higher than 4 corresponds to an "Excellent" perceived quality. Since it is very important to provide the best QoE as possible to the users, the scheme comparison is also performed for S = 4. Figures 5.4(a) and 5.4(b) plot the percentage of very satisfied users (S = 4), for all schemes, with a latency of 10 ms and 1 ms, respectively.

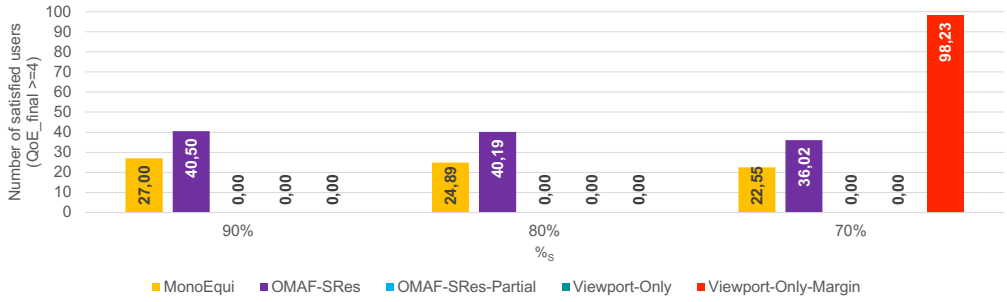


(a) L = 10 ms (b) L = 1 ms

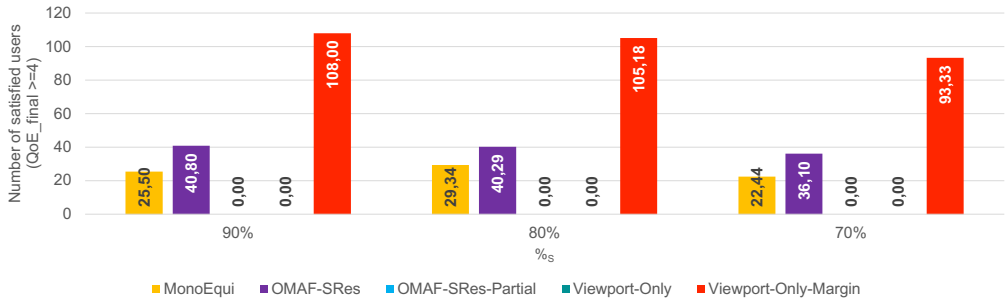
Figure 5.4: Percentage of satisfied users (S = 4) for all schemes.

Figures 5.5(a) and 5.5(b) present the number of very satisfied users (S = 4), for all schemes, with a latency of 10 ms and 1 ms, respectively.

According to Figure 5.5, the Viewport-Only-Margin scheme does not work with this high level of user satisfaction with a latency of 10 ms, except when the percentage of satisfied users is 70%. For S = 4, the performance gains of OMAF-SRes over MonoEqui are also higher across all percentages of satisfied users and latency values. The number of satisfied users with these two schemes are similar for both



(a) L = 10 ms



(b) L = 1 ms

Figure 5.5: Number of satisfied users (S = 4) for all schemes, $\%_S$ = 90%, 80% and 70%.

latency values. Also, the Viewport-Only scheme does not work with S = 4.

Tables 5.14 and 5.15 present for each scheme the number of very satisfied users (S = 4), the total number of connected users and the capacity gains relative to MonoEqui, for S = 4 and $\%_S$ = 90%, 80% and 70%, for a latency of 10 ms and 1 ms, respectively.

Table 5.14: Capacity of all schemes for S = 4, $\%_S$ = 90%, 80% and 70%. L = 10 ms.

$\%_S$	Scheme (Satisfied users/Connected users/Gain)				
	MonoEqui	**OMAF-SRes**	**OMAF-SRes-Partial**	**Viewport-Only**	**Viewport-Only-Margin**
90%	**27.00** 30.00 (0.00%)	**40.50** 45.00 (+50.00%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)
80%	**24.89** 31.11 (0.00%)	**40.19** 50.24 (+61.47%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)
70%	**22.55** 32.22 (0.00%)	**36.02** 51.46 (+59.73%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**98.23** 140.33 (+335.61%)

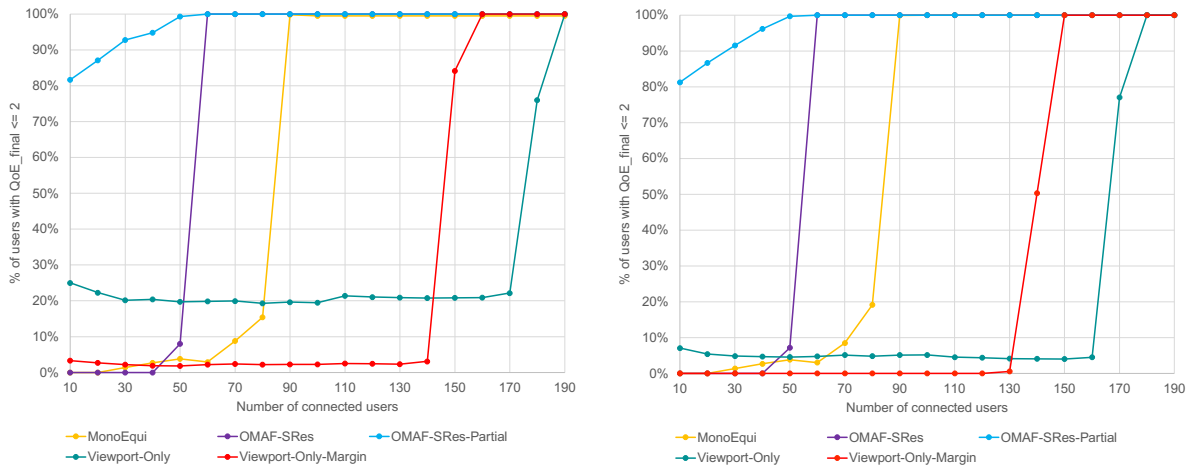
5.4.3 Comparison of the Number of Non-Satisfied Users (NS = 2)

Just as it is important to compare the number of satisfied users of the various schemes, it is also important to compare the number of non-satisfied users. Figures 5.6(a) and 5.6(b) plot the percentage

Table 5.15: Capacity of all schemes for S = 4, $\%_S$ = 90%, 80% and 70%. L = 1 ms.

$\%_S$	Scheme (Satisfied users/Connected users/Gain)				
	MonoEqui	**OMAF-SRes**	**OMAFSRes-Partial**	**Viewport-Only**	**Viewport-Only-Margin**
90%	**25.50** 28.33 (0.00%)	**40.80** 45.33 (+60.00%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**108.00** 120.00 (+323.53%)
80%	**29.34** 36.67 (0.00%)	**40.29** 50.36 (+37.32%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**105.18** 131.48 (+258.49%)
70%	**22.44** 32.05 (0.00%)	**36.10** 51.57 (+60,87%)	**0.00** 0.00 (-100.00%)	**0.00** 0.00 (-100.00%)	**93.33** 133.33 (+315.91%)

of non-satisfied users (NS = 2), for all schemes, with a latency of 10 ms and 1 ms, respectively.



(a) L = 10 ms (b) L = 1 ms

Figure 5.6: Percentage of non-satisfied users (NS = 2) for all schemes.

The comparison of the various schemes for NS = 2 is done as a function of the number of very satisfied users (S = 4). As such, Tables 5.16 and 5.17 present for each scheme the number of connected users for S = 4, $\%_S$ = 90%, 80% and 70%, the corresponding number of non-satisfied users, and the corresponding $\%_{NS}$ with NS = 2, for a latency of 10 ms and 1 ms, respectively. As it is possible to observe, some values cannot be computed because the corresponding capacity in terms of very satisfied users with S = 4 is zero. For a latency of 10 ms, the scheme that presents the lowest numbers of non-satisfied users is MonoEqui. For OMAF-SRes, the number of non-satisfied users is low for when the number of connected users is the one that supports S = 4, $\%_S$ = 90% and 80%. For 70%, the number of non-satisfied users is much higher for only 1.22 more connected users. For a latency of 1 ms, the number of non-satisfied users is comparable to the latency value of 10 ms for MonoEqui and OMAF-SRes. For Viewport-Only-Margin, although the number of non-satisfied users when the number of connected users is the one that supports S = 4, $\%_S$ = 70% is much higher, when $\%_S$ = 90%, the

number of non-satisfied users is zero, which is the only case where this happens.

Table 5.16: Non-satisfied users corresponding to S = 4, $\%_S$ = 90%, 80% and 70%. L = 10 ms.

$\%_S$	Scheme (Connected users/$\%_{NS}$/Non-satisfied users)				
	MonoEqui	**OMAF-SRes**	**OMAFSRes-Partial**	**Viewport-Only**	**Viewport-Only-Margin**
90%	30.00 1.00% **0.30**	45.00 4.00% **1.8**	N/A	N/A	N/A
80%	31.11 1.22% **0.38**	50.24 10.21% **5.13**	N/A	N/A	N/A
70%	32.22 1.44% **0.46**	51.46 21.43% **11.03**	N/A	N/A	140.33 5.67% **7.96**

Table 5.17: Non-satisfied users corresponding to S = 4, $\%_S$ = 90%, 80% and 70%. L = 1 ms.

$\%_S$	Scheme (Connected users/$\%_{NS}$/Non-satisfied users)				
	MonoEqui	**OMAF-SRes**	**OMAFSRes-Partial**	**Viewport-Only**	**Viewport-Only-Margin**
90%	28.33 0.83% **0.24**	45.33 3.73% **1.69**	N/A	N/A	120.00 **0** 0.00%
80%	36.67 2.33% **0.85**	50.36 10.35% **5.21**	N/A	N/A	131.48 8.25% **10.85**
70%	32.05 1.41% **0.45**	51.57 21.6% **11.14**	N/A	N/A	133.33 17.32% **23.09**

5.5 Results Summary and Analysis

Based on Tables 5.12 to 5.17, this section presents the most relevant results and conclusions:

- In a scenario where MEC infrastructure is not deployed, the OMAF-SRes scheme is better than MonoEqui at satisfying users, especially when aiming at high satisfaction levels. It achieves 40.50 very satisfied users for $\%_S$ = 90%, which is 50% more than MonoEqui. The only drawback is that the corresponding number of non-satisfied users will be 1.69 instead of 0.24. However, it is considered that the relative improvement in very satisfied users is enough to justify this drawback.
- When considering the MonoEqui and OMAF-SRes schemes, the gains of deploying MEC infrastructure are very small, if existent at all. As previously discussed, a latency reduction in the order of 10 ms has a residual impact when the buffer threshold is in the order of hundreds of milliseconds.

- The OMAF-SRes-Partial scheme should not be considered since it does not support any satisfied or very satisfied users for any $\%_S$. As discussed previously, the fact that the video area is small and the segments are 1 second long makes this an unsuitable scheme.
- The Viewport-Only scheme does not work without MEC infrastructure. With MEC infrastructure deployed, it can only achieve 80% of satisfied users (S = 3). However, at this level, it is possible to achieve 128.26 satisfied users, which is the most out of any scenario. This corresponds to a total of 160.32 connected users and 11.7 non-satisfied users ($\%_{NS}$ = 7.3%).
- Although the Viewport-Only-Margin supports around 100 satisfied users (S = 3) for both latency values, when considering very satisfied users (S = 4), it only supports 70% of users when there is no MEC infrastructure, with 7.96 non-satisfied users. The only way for Viewport-Only-Margin to support 90% of users with an ”Excellent” perceived quality is to deploy MEC infrastructure. If MEC infrastructure is present, not only is it possible to support 108 very satisfied users, the highest value for S = 4, it is also possible to achieve zero non-satisfied users, being the only configuration that achieves this.

Chapter 6

Conclusions

This work aimed at studying how different 360° video streaming schemes compared against each other in terms of their performance and resulting user satisfaction. Another goal was to study how Multi-Access Edge Computing (MEC) may be beneficial for these types of applications.

Drawing from previous work and by developing the simulator described in Chapter 4, it was possible to assess the performance, in terms of perceived quality by the users, of 5 different delivery schemes, MonoEqui, OMAF-SRes, OMAF-SRes-Partial, Viewport-Only and Viewport-Only-Margin, for two different latency levels, 10 ms and 1 ms, representing the deployment or not of MEC infrastructure, for various numbers of connected users and for various levels of satisfaction and non-satisfaction. This was done by considering different varying channel conditions for each user and realistic viewport trajectories and video bitrates, while employing a QoE model to assess the perceived quality by the user. A sensitivity analysis was then performed in order to estimate the best settings for each scheme, considering the statistical relevancy of the results.

It was found that, when no MEC infrastructure is deployed, the OMAF-SRes is a better streaming scheme than the commonly used MonoEqui, supporting 40.50 very satisfied users while keeping the number of non-satisfied users fairly low. When the 5G network is leveraged by MEC servers, making a remote viewport rendering solution viable, 120 very satisfied users can be supported with Viewport-Only-Margin, which is three times better than what OMAF-SRes can achieve without MEC, while achieving zero non-satisfied users. While it is true that Viewport-Only and Viewport-Only-Margin can support high numbers of satisfied users, it is important to remember that today's paradigm of high quality multimedia streaming applications means that users are extremely demanding in terms of the quality of experience they receive from the services they pay for. Furthermore, users are particularly sensitive to 360° video, suffering from motion sickness and other effects when the quality is below standard. Therefore, mobile network operators and other stakeholders must aim to deliver QoEs higher than 3. Taking this into account, it is concluded that there is a clear advantage to using MEC infrastructure in 5G networks in order to leverage 360° video streaming.

It is also important to take into account that the performed simulations were made considering a fairly modest configuration in comparison to what 5G networks are expected to achieve in the future. The maximum achievable throughput with the studied network was 170.1 Mbps, while 5G networks are expected to be able to achieve 20 Gbps of peak data rates. This will unlock even higher capacities for this and other types of applications and support even higher 360° video quality with 8K stereoscopic

resolutions and 120 frames per second.

One drawback from this work is that only the end-to-end network latency is considered. In order to be more realistic, other latency components should be considered, such as the latencies associated with the HMD responding to a user movement, processing the new viewport, decoding video and refreshing the display. Considering a video with 60 frames per second this motion-to-update latency must be no higher than 16 ms [50].

Although the relative performance of different QoE schemes is not affected, one other limitation of this work is the lack of formal research into QoE models for 360° video. As a consequence, a 2D video QoE model had to be adapted to fit the 360° video use case with assumptions about how users perceive these videos.

For future work, it would be interesting to perform subjective studies on the perceived quality of 360° videos using the different delivery schemes in order to develop more accurate QoE models. The simulator could be further developed in order to include new capabilities such as coexisting with other latency-critical services that compete in terms of prioritization with the 360° video delivery or having users give up from their streaming session if their QoE stays too low for more than a certain amount of time. It would also be interesting to study how mobile network operators can operationalize the deployment of MEC infrastructure in terms of location and number of servers in order to take advantage of the performance gains while keeping their Capital Expenditures (CAPEX) and Operational Expenditures (OPEX) at feasible levels.

Bibliography

[1] Ericsson. *Ericsson Mobility Report*. Ericsson, November 2020. Accessed on: July 20, 2020. [Online]. Available: https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf.

[2] Huawei. *Preparing for a Cloud AR/VR Future*. Huawei. Accessed on: July 20, 2020. [Online]. Available: cloud_vr_ar_white_paper_en.pdf:/Users/miguel/Zotero/storage/RG749X45/cloud_vr_ar_white_paper_en.pdf:application/pdf.

[3] Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022*. Cisco, February 2019. Accessed on: December 13, 2020. [Online]. Available: https://s3.amazonaws.com/media.mediapost.com/uploads/CiscoForecast.pdf.

[4] M. Zink, R. Sitaraman, and K. Nahrstedt. Scalable 360° Video Stream Delivery: Challenges, Solutions, and Opportunities. *Proceedings of the IEEE, vol. 107, no. 4, pp. 639-650*, April 2019. doi: 10.1109/JPROC.2019.2894817.

[5] H. Holma, A. Toskala, and T. Nakamura, editors. *5G technology: 3GPP new radio*. Wiley, Hoboken, NJ, first edition, 2020. ISBN 978-1-119-23629-0 978-1-119-23628-3.

[6] ITU-R. *Recommendation ITU-R M.2083-0, IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond*. ITU-R, 2015. Accessed on: November 7, 2020. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf.

[7] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, and G. Wunder. 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE Journal on Selected Areas in Communications, vol. 35, no.6, pp. 1201-1221*, June 2017. doi: 10.1109/JSAC.2017.2692307.

[8] C. Song, M. Zhang, Y. Zhan, D. Wang, L. Guan, W. Liu, L. Zhang, and S. Xu. Hierarchical edge cloud enabling network slicing for 5G optical fronthaul. *IEEE/OSA Journal of Optical Communications and Networking, vol. 11, no. 4, pp. B60-B70*, April 2019. doi: 10.1364/JOCN.11.000B60.

[9] Q. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W. Hwang, and Z. Ding. A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art. *IEEE Access, vol. 8, pp. 116974-117017*, June 2020. doi: 10.1109/ACCESS.2020.3001277.

[10] 3GPP. *Release 15 Description: Summary of Rel-15 Work Items*. 3GPP, October 2019. Technical Report (TR) 21.915, version 15.0.0. Accessed on: November 9, 2020. [Online]. Available: https://www.3gpp.org/release-15.

[11] Cisco. *5G Non Standalone Solution Overview*. Cisco. Accessed on: November 12, 2020. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21-5_N5-8/5G-NSA/21-5-5G-NSA-Solution-Guide/21-5-5G-NSA-Solutions-Guide_chapter_01.pdf.

[12] GSMA. *Road to 5G: Introduction and Migration*. GSMA, April 2018. Accessed on: November 13, 2020. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2018/04/Road-to-5G-Introduction-and-Migration_FINAL.pdf.

[13] H. Menendez. *Ethernet for 5G fronthaul*. Nokia, April 2019. Accessed on: November 24, 2020. [Online]. Available: https://www.nokia.com/blog/ethernet-5g-fronthaul/.

[14] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. Cloud RAN for Mobile Networks - A Technology Overview. *IEEE Communications Surveys and Tutorials, vol. 17, no. 1, pp. 405-426*, First quarter 2015. doi: 10.1109/COMST.2014.2355255.

[15] J. Wu, Z. Zhang, Y. Hong, and Y. Wen. Cloud Radio Access Network (C-RAN): A Primer. *IEEE Network, vol. 29, no. 1, pp. 35-41*, January 2015. doi: 10.1109/MNET.2015.7018201.

[16] 5G-PPP. *View on 5G Architecture*. 5G-PPP, December 2017. Accessed on: November 25, 2020. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf.

[17] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati. Network function virtualization in 5G. *IEEE Communications Magazine, vol. 54, no. 4, pp. 84-91*, April 2016. doi: 10.1109/MCOM.2016.7452271.

[18] GTI. *GTI Sub-6GHz 5G Device White Paper*. GTI, January 2018. Accessed on: November 13, 2020. [Online]. Available: http://gtigroup.org/d/file/Resources/rep/2018-02-22/c9d31709d72643e625321d6f3724d761.pdf.

[19] B. Bertenyi, R. Burbidge, G. Masini, S. Sirotkin, and Y. Gao. NG Radio Access Network (NG-RAN). *Journal of ICT Standardization, vol. 6, no. 1, pp. 59-76*, 2018. doi: 10.13052/jicts2245-800X.614.

[20] 3GPP. *Base Station (BS) radio transmission and reception*. 3GPP, June 2019. Technical Specification (TS) 38.104, version 15.6.0. Accessed on: November 28, 2020. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3202.

[21] A. A. Zaidi, R. Baldemair, V. Moles-Cases, N. He, K. Werner, and A. Cedergren. OFDM Numerology Design for 5G New Radio to Support IoT, eMBB, and MBSFN. *IEEE Communications Standards Magazine, vol. 2, no. 2, pp. 78-83*, June 2018. 10.1109/MCOMSTD.2018.1700021.

[22] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. D. Silva. Waveform and Numerology to Support 5G Services and Requirements. *IEEE Communications Magazine, vol. 54, no. 11, pp. 90-98*, November 2016. 10.1109/MCOM.2016.1600336CM.

[23] J. Campos. *Understanding the 5G NR Physical Layer*. Keysight Technologies, November 2017. Accessed on: November 28, 2020. [Online]. Available: https://www.keysight.com/upload/cmc_upload/All/Understanding_the_5G_NR_Physical_Layer.pdf.

[24] 3GPP. *Physical channels and modulation*. 3GPP, June 2019. Technical Specification (TS) 38.211, version 15.6.0. Accessed on: November 28, 2020. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3213.

[25] Qualcomm. *Designing 5G NR*. Qualcomm, September 2018. Accessed on: November 28, 2020. [Online]. Available: https://www.qualcomm.com/media/documents/files/the-3gpp-release-15-5g-nr-design.pdf.

[26] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim. Ultra-Reliable and Low-Latency Communications in 5G Downlink: Physical Layer Aspects. *IEEE Wireless Communications, vol. 25, no. 3, pp. 124-130*, June 2018. doi: 10.1109/MWC.2018.1700294.

[27] S. Ahmadi. *5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards*. Elsevier Science, Cambridge, 2019. ISBN 978-0-08-102267-2.

[28] 3GPP. *User Equipment (UE) radio access capabilities*. 3GPP, October 2018. Technical Specification (TS) 38.306, version 15.3.0. Accessed on: June 27, 2021. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3193.

[29] 3GPP. *Physical layer procedures for data*. 3GPP, October 2018. Technical Specification (TS) 38.214, version 15.3.0. Accessed on: June 28, 2021. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216.

[30] 5G NR Modulation and Coding Scheme - Modulation and Code Rate, July 2020. Accessed on: June 28, 2021. [Online]. Available: https://www.techplayon.com/5g-nr-modulation-and-coding-scheme-modulation-and-code-rate/.

[31] S. S. Hadi and T. C. Tiong. Adaptive Modulation and Coding for LTE Wireless Communication. *IOP Conference Series: Materials Science and Engineering, vol. 78, pp. 012016*, April 2015. doi: 10.1088/1757-899X/78/1/012016.

[32] Rohde & Schwarz. *R&S TS8980 test system analyzes LTE quality indicators: CQI, PMI and RI*. Rohde & Schwarz. Accessed on: July 28, 2021. [Online]. Available: https://cdn.rohde-schwarz.com/magazine/pdfs_1/article/203/NEWS_203_english_TS8980.pdf.

[33] 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*, February 2018. doi: 10.1109/JIOT.2017.2750180.

[34] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva. VR is on the Edge: How to Deliver 360° Videos in Mobile Networks. In *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network - VR/AR Network '17*, August 2017. doi: 10.1145/3097895.3097901.

[35] ETSI. *Mobile-edge computing introductory technical white paper*. ETSI, September 2014. Accessed on: November 29, 2020. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf.

[36] A. Yaqoob, T. Bi, and G. Muntean. A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities. *IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2801-2838*, 2020. doi: 10.1109/COMST.2020.3006999.

[37] M. P. Queluz, F. Lopes, J. Ascenso, and A. Rodrigues. Subjective and objective quality assessment of omnidirectional video. In *Applications of Digital Image Processing XLI*, San Diego, CA, September 2018. SPIE. doi: 10.1117/12.2321679.

[38] D. Podborski, E. Thomas, M. M. Hannuksela, S. Oh, T. Stockhammer, and S. Pham. Virtual Reality and DASH. In *IBC Conference*, Amsterdam, 2017.

[39] M. M. Hannuksela, Y. Wang, and A. Hourunranta. An Overview of the OMAF Standard for 360° Video. In *2019 Data Compression Conference (DCC)*, March 2019. 10.1109/DCC.2019.00050.

[40] R. Shafi, W. Shuai, and M. U. Younus. 360-Degree Video Streaming: A Survey of the State of the Art. *Symmetry, vol. 12, no. 9, pp. 1491*, September .

[41] J. Evans. *MPEG Encoding and IPTV SLA Requirements*. Cisco, December 2008. Accessed on: June 28, 2021. [Online]. Available: https://www.cisco.com/web/offer/apj/asiatechforum/Post_Event/Day1-4_Dec/WorkshopD2-JohnEvans-MPEG_Encoding_and_IPTV_SLA_Requirements.pdf.

[42] K. Vijayanagar. *I, P, and B-frames - Differences and Use Cases Made Easy*, December 2020. Accessed on: July 28, 2021. [Online]. Available: https://ottverse.com/i-p-b-frames-idr-keyframes-differences-usecases/.

[43] Y. Xu, S. Xie, Q. Shen, Z. Ma, I. Bouazizi, and Y. Wang. Omnidirectional Media Format and Its Application to Immersive Video Streaming: An Overview. 2018.

[44] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia, vol. 18, no. 4, pp. 62-67*, April 2011. doi: 10.1109/MMUL.2011.71.

[45] T. Stockhammer. Dynamic adaptive streaming over HTTP: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems*, San Jose, CA, USA, February 2011. ACM. doi: 10.1145/1943552.1943572.

[46] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. Viewport-adaptive navigable 360-degree video delivery. In *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017. IEEE. doi: 10.1109/ICC.2017.7996611.

[47] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim. MPEG DASH SRD: spatial relationship description. In *Proceedings of the 7th International Conference on Multimedia Systems*, Klagenfurt, Austria, May 2016. ACM. doi: 10.1145/2910017.2910606.

[48] C. Zhou, Z. Li, and Y. Liu. A Measurement Study of Oculus 360 Degree Video Streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, June 2017. ACM. doi: 10.1145/3083187.3083190.

[49] H. Koumaras, M.-A. Kourtis, and C. Skianis. Compression Performance and Video Quality Comparison of HEVC and AVC. In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, Greece, December 2014. IEEE. doi: 10.1109/CAMAD.2014.7033209.

[50] S. Shi, V. Gupta, M. Hwang, and R. Jana. Mobile VR on Edge Cloud: A Latency-Driven Design. In *Proceedings of the 10th ACM Multimedia Systems Conference*, Amherst, Massachusetts, June 2019. ACM. doi: 10.1145/3304109.3306217.

[51] *ISO/IEC 23090-2:2019 - Information technology - Coded representation of immersive media - Part 2: Omnidirectional media format*. First edition, January 2019.

[52] M. Luís. Viewport Adaptive Streaming for Omnidirectional Video Delivery. MSc Thesis, Instituto Superior Técnico, Lisbon, Portugal, June 2020.

[53] K. Brunnström, S. A. Beker, K. de Moor, A. Dooms, and S. Egger. Qualinet White Paper on Definitions of Quality of Experience. 2013.

[54] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang. 5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks. *IEEE Transactions on Broadcasting, vol. 64, no. 2, pp. 621-634*, June 2018. doi: 10.1109/TBC.2018.2816786.

[55] Z. Akhtar and T. H. Falk. Audio-Visual Multimedia Quality Assessment: A Comprehensive Survey. *IEEE Access, vol. 5, pp. 21090-21117*, September 2017. doi: 10.1109/TBC.2018.2816786.

[56] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys Tutorials, vol. 17, no. 1, pp. 469-492*, 2015. doi: 10.1109/COMST.2014.2360940.

[57] M. Yu, H. Lakshman, and B. Girod. A Framework to Evaluate Omnidirectional Video Coding Schemes. In *2015 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, September 2015. doi: 10.1109/ISMAR.2015.12.

[58] M. Hu, X. Luo, J. Chen, Y. C. Lee, Y. Zhou, and D. Wu. Virtual reality: A survey of enabling technologies and its applications in IoT. *Journal of Network and Computer Applications, vol. 178, pp. 102970*, March 2021. doi: 10.1016/j.jnca.2020.102970.

[59] A. Horé and D. Ziou. Image Quality Metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*. IEEE, August 2010. doi: 10.1109/ICPR.2010.579.

[60] X. Hu, W. Quan, T. Guo, Y. Liu, and L. Zhang. Mobile Edge Assisted Live Streaming System for Omnidirectional Video. *Mobile Information Systems, vol. 2019, pp. 1-15*, May 2019. doi: 10.1155/2019/8487372.

[61] Y. Bao, T. Zhang, A. Pande, H. Wu, and X. Liu. Motion-Prediction-Based Multicast for 360-Degree Video Transmissions. In *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, San Diego, CA, USA, June 2017. IEEE. doi: 10.1109/SAHCN.2017.7964928.

[62] F. Rodrigues, I. Sousa, M. P. Queluz, and A. Rodrigues. QoE-Aware Scheduling Algorithm for Adaptive HTTP Video Delivery in Wireless Networks. *Wireless Communications and Mobile Computing, vol. 2018, pp. e9736360*, September 2018. doi: 10.1155/2018/9736360.

[63] D. Suh, I. Jang, and S. Pack. QoE-enhanced adaptation algorithm over DASH for multimedia streaming. In *The International Conference on Information Networking 2014 (ICOIN2014)*, Phuket, Thailand, February 2014. IEEE. doi: 10.1109/ICOIN.2014.6799731.

[64] T. Thang, Q.-D. Ho, J. Kang, and A. Pham. Adaptive streaming of audiovisual content using MPEG DASH. *IEEE Transactions on Consumer Electronics, vol. 58, no. 1, pp. 78-85*, February 2012. doi: 10.1109/TCE.2012.6170058.

[65] J. Boyce, E. Alshina, A. Abbas, and Y. Ye. *JVET-H1030: JVET common test conditions and evaluation procedures for 360º video*. JVET, July 2018.

[66] H. Koumaras, M.-A. Kourtis, and C. Skianis. Compression performance and video quality comparison of HEVC and AVC. In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, Greece, December 2014. IEEE. doi: 10.1109/CAMAD.2014.7033209.

[67] F. Capozzi, G. Piro, L. Grieco, G. Boggia, and P. Camarda. Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey. *IEEE Communications Surveys & Tutorials, vol. 15, no. 2, pp. 678-700*, 2013. doi: 10.1109/SURV.2012.060912.00100.

[68] M. K. Muller, S. Schwarz, and M. Rupp. QoS investigation of proportional fair scheduling in LTE networks. In *2013 IFIP Wireless Days (WD)*, Valencia, Spain, November 2013. IEEE. doi: 10.1109/WD.2013.6686478.

[69] I. Sousa, M. P. Queluz, and A. Rodrigues. A survey on QoE-oriented wireless resources scheduling. *Journal of Network and Computer Applications, vol. 158, pp. 102594*, May 2020. doi: 10.1016/j.jnca.2020.102594.

[70] B. O'Donnell. *5G Latency Improvements Are Still Lagging*. Forbes, February 2020. Accessed on: July 17, 2021. [Online]. Available: `https://www.forbes.com/sites/bobodonnell/2020/02/18/5g-latency-improvements-are-still-lagging/`.

[71] G. Brown. *New Transport Network Architectures for 5G RAN*. Fujitsu. Accessed on: July 17, 2021. [Online]. Available: `https://www.fujitsu.com/us/Images/New-Transport-Network-Architectures-for-5G-RAN.pdf`.

[72] C. Wu, Z. Tan, Z. Wang, and S. Yang. A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, Taipei, Taiwan, June 2017. ACM. doi: 10.1145/3083187.3083210.

[73] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck. QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 12, no. 2, pp. 1-24*, March 2016. doi: 10.1145/2818361.

[74] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck. Design and optimisation of a (FA)Q -learning-based HTTP adaptive streaming client. *Connection Science, vol. 26, no. 1, pp. 25-43*, January 2014. doi: 10.1080/09540091.2014.885273.

[75] J. De Vriendt, D. De Vleeschauwer, and D. Robinson. Model for estimating QoE of video delivered using HTTP adaptive streaming. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, Ghent, Belgium, 2013. IEEE. ISBN: 978-3-901882-50-0.

[76] J. Carneiro. *A Tutorial on the Central Limit Theorem*. Accessed on: September 29, 2021. [Online]. Available: `http://qobweb.igc.gulbenkian.pt/demos/clt/`.

[77] M. D. Byrne. How Many Times Should a Stochastic Model Be Run? An Approach Based on Confidence Intervals.

[78] R. Bandeira. Analysis of Spectrum Sharing Techniques for Unlicensed Frequency Bands. MSc Thesis, Instituto Superior Técnico, Lisbon, Portugal, May 2017.

Appendix A

QAAD Algorithm Implementation

A.1 Bitrate Selection

One of the objectives of QAAD is to minimize the number of quality changes. The bitrate selection scheme takes into account the current buffer level of the client as well as the available throughput. Each time the algorithm is called, it starts by determining the best quality it is possible to achieve, l_{best}, by taking into account the highest bitrate it is possible to request without surpassing the estimated available bandwidth, $bw_{estimated}$. The original QAAD pseudocode is presented in Algorithm 2.

Algorithm 2 Pseudocode of the bitrate selection scheme in QAAD [63].

1: **if** $l_{best} == l_{prev}$ **then**
2: $\quad l_{next} = l_{prev}$
3: **else if** $l_{best} > l_{prev}$ **then**
4: $\quad$ **if** $B > \mu$ **then**
5: $\quad\quad l_{next} = l_{prev} + 1$
6: $\quad$ **else**
7: $\quad\quad l_{next} = l_{prev}$
8: $\quad$ **end if**
9: **else if** $l_{best} < l_{prev}$ **then**
10: $\quad k = 0$
11: $\quad$ **do**
12: $\quad\quad t_{l_{prev}-k,\sigma} = \frac{B(t)-\sigma}{1-\frac{bw_{estimated}}{b(l_{prev}-k)}}$
13: $\quad\quad n_{l_{prev}-k} = \frac{t_{l_{prev}-k,\sigma} \cdot bw_{estimated}}{\tau \cdot b(l_{prev}-k)}$
14: $\quad\quad k = k + 1$
15: $\quad$ **while** $n_{l_{prev}-k} < 1$ **and** $k < l_{prev} - 1$
16: $\quad l_{next} = l_{prev} - k$
17: **end if**

The algorithm tests three cases. For the first case ($l_{best} == l_{prev}$, line 1), no quality change is necessary because the best possible bitrate is already being requested. As such, l_{next} is simply determined as the previously requested quality, l_{prev} (line 2).

If the second case ($l_{best} > l_{prev}$, line 3) is true, it means that it is possible to increase the requested quality. However, this only happens if the current buffer level, B, is higher than a pre-defined marginal buffer length, μ (line 4). This step is taken in order to protect the buffer from unexpected throughput fluctuations. If the current buffer level is below the marginal buffer length, l_{next} is set to mantain the previously requested quality, l_{prev} (line 7).

If the third case ($l_{best} < l_{prev}$, line 9) is true, it means that the current bandwidth cannot support the previously requested quality, and therefore the requested quality should be reduced. However, being too aggressive in the quality reduction would have a significant impact on the user's QoE. In order to avoid this, the QAAD tries to keep the next quality level comparable to the previous one. In order to do this, the algorithm calculates two values: the remaining playback time left in the buffer, $t_{l,\sigma}$, that preserves a pre-defined minimal buffer length, σ, and the expected number of segments that can be downloaded during that time if the estimated bandwidth remains constant, n_l, as

$$t_{l,\sigma} = \frac{B\left(t\right) - \sigma}{1 - \frac{bw_{estimated}}{b(l)}}, \tag{A.1a}$$

$$n_l = \frac{t_{l,\sigma} \cdot bw_{estimated}}{\tau \cdot b(l)}, \tag{A.1b}$$

where the parameters $b(l)$ and τ are the bitrate of the quality level l and the segment duration, respectively. The deduction of these metrics can be found in [63].

Naturally, the values of these metrics are dependent on the bitrate of the requested quality, $b(l)$. Therefore, the algorithm must search for the maximum feasible quality level (equal to or lower than l_{prev}) that produces $n_l > 1$. At this quality level, the algorithm expects to download the whole segment in, at most, the time it takes the buffer to drop to the pre-defined minimum level. This quality level is found by iteratively incrementing the index k and calculating these metrics for the associated quality level (lines 11-15). If no feasible quality level exists, k reaches its maximum level and the lowest quality level is selected.

It is important to note that, in the case of $l_{best} < l_{prev}$, according to line 14 in Algorithm 2, k will always be equal to or greater than 1, as k is always incremented at least once. Therefore, the requested quality of the new segment will always decrease at least one level ($l_{next} = l_{prev} - k$, line 16). This goes against the stated strategy of minimizing the number of decreases in quality and is incoherent with what is presented in the paper's results. As such, a modification was made to the algorithm in order to reflect the proposed behavior. More specifically, line 16 was modified to $l_{next} = l_{prev} - (k-1)$. This modification requires the second condition in line 15 to be modified from $k < l_{prev} - 1$ to $k < l_{prev}$, in order to prevent the value of l_{next} to be set to less than the minimum quality level. This way, it becomes possible for the requested quality level to remain the same when the algorithm enters this condition.

An additional modification had to be made to the algorithm, taking into account the fact that there is a possibility that the calculation of $t_{l_{prev}-k,\sigma}$ in line 12 is negative, which may sometimes cause the algorithm to work in unexpected ways. From (A.1a), it is possible to deduce that the value of $t_{l_{prev}-k,\sigma}$ can be negative in two situations: when the client's buffer level, B, is lower than the predefined minimal buffer level, σ, or when the estimated available throughput is larger than the bitrate of the quality level that the algorithm is testing.

In the first case, this means that the requested quality will need to be lowered to a value equal to or below the estimated throughput. Even though $n_{l_{prev}-k}$ starts by being negative due to $t_{l_{prev}-k,\sigma}$ also being negative, this value starts to grow as k is being incremented with the objective of reaching a value

greater than 1. This is normal behavior.

The value of $t_{l_{prev}-k,\sigma}$ may also be negative due to the estimated throughput being greater than the bitrate of the quality level. When k keeps being incremented to a value high enough, the bitrates of the quality levels start being smaller than the estimated bandwidth, making $t_{l_{prev}-k,\sigma}$, which was growing with each increment of k, to suddenly turn negative. This happens because, when the requested bitrate becomes lower than the estimated bandwidth, the buffer starts growing, meaning it never depletes. In fact, the authors in [63] state that $b(l)$ must be greater than $bw_{estimated}$ for (A.1a) to be valid. However, the pseudocode does not contemplate the eventuality that this condition is not met. If the pseudocode is not modified, what happens is that every time this situation occurs, the requested quality level will always be the minimum possible, because the condition $n_{l_{prev}-k} \geq 1$ (line 15) is never verified.

In order to illustrate this problem, an experiment is conducted where the available bitrates are: $\{$400, 500, 600, 700, 800, 1000, 1200, 1500, 2000$\}$ kbps. It is assumed that the available bandwidth is 2200 kbps and therefore that the client is in a state where it is requesting the maximum quality level. If the available bandwidth suddenly drops to 800 kbps, the client keeps requesting the maximum quality level up until the moment it enters the rate adaptation algorithm and determines it cannot download the next segment in the time it takes the buffer to deplete do the minimal level if the maximum quality level is maintained. When this happens, the client needs to start decreasing the requested quality in order to avoid a stall and therefore enters the loop in lines 11-15 of Algorithm 2. Table A.1 illustrates the calculated values inside this loop as k is incremented, assuming that the minimal buffer level is set to 3 seconds and the buffer level at the time the algorithm is called is 3.236 seconds.

Table A.1: QAAD calculations for incremental values of k.

k	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$b(l_{prev}-k)$ [kbps]	2000	1500	1200	1000	810	800	790	700	600	500
$t_{l_{prev}-k,\sigma}$ [s]	0.393	0.506	0.708	1.18	19.116	∞	-18.644	-1.652	-0.708	-0.393
$n_{l_{prev}-k}$	0.079	0.135	0.236	0.472	9.44	∞	-9.44	-0.944	-0,472	-0.315

In this case, the requested quality level for the next segment will be the one corresponding to a bitrate of 810 kbps, as it is the first one that meets the condition $n_{l_{prev}-k} \geq 1$, which breaks the loop. In this case, the selected bitrate is slightly higher than the available throughput, meaning that the buffer would keep depleting at a slower pace. If the bandwidth remains constant, the quality level would need to be further lowered at some point in the future, but this is expected behavior.

Considering now a scenario where this bitrate is not available in the server, the quality level corresponding to 800 kbps would be chosen, as it also meets the condition $n_{l_{prev}-k} \geq 1$. With this quality level selected, the bitrate would equal the available throughput and the buffer would stay stable until the end of the session. As it can be observed in Table A.1, this bitrate corresponds to the limit scenario where the algorithm behaves as expected.

In a scenario where this quality level is also not available, it would make sense for the algorithm to select the next quality level (i.e., 790 kbps). With this quality level selected, the buffer would start slowly

increasing, since the available throughput is slightly higher than the bitrate. However, because the buffer would start increasing, $t_{l_{prev}-k,\sigma}$ would become negative, making $n_{l_{prev}-k}$ also negative, meaning the condition $n_{l_{prev}-k} \geq 1$ not met. Therefore, the loop would keep iterating k until reaching the lowest quality level, impacting the quality further than necessary.

In a real scenario, it is not possible to count on one of the available bitrates being slightly lower than or exactly equal to the available throughput, as this happens only by chance. Therefore, an adaptation must be made to the algorithm in order to prevent the presented situation. Below line 12 in Algorithm 2, a condition was put in place to increment k and break the loop if $t_{l_{prev}-k,\sigma} < 0$ and $B(t) - \sigma > 0$.

The adapted QAAD pseudocode is presented in Algorithm 3.

Algorithm 3 Modified pseudocode of the bitrate selection scheme in QAAD [63].

1: **if** $l_{best} == l_{prev}$ **then**
2: $\quad l_{next} = l_{prev}$
3: **else if** $l_{best} > l_{prev}$ **then**
4: $\quad$ **if** $B > \mu$ **then**
5: $\quad\quad l_{next} = l_{prev} + 1$
6: $\quad$ **else**
7: $\quad\quad l_{next} = l_{prev}$
8: $\quad$ **end if**
9: **else if** $l_{best} < l_{prev}$ **then**
10: $\quad k = 0$
11: $\quad$ **do**
12: $\quad\quad t_{l_{prev}-k,\sigma} = \frac{B(t)-\sigma}{1-\frac{bw_{estimated}}{b(l_{prev}-k)}}$
13: $\quad\quad$ **if** $t_{l_{prev}-k,\sigma} < 0$ **and** $B\,(t) - \sigma > 0$ **then** ▷ Modification
14: $\quad\quad\quad k = k + 1$ ▷ Modification
15: $\quad\quad\quad$ **break** ▷ Modification
16: $\quad\quad$ **end if** ▷ Modification
17: $\quad\quad n_{l_{prev}-k} = \frac{t_{l_{prev}-k,\sigma} \cdot bw_{estimated}}{\tau \cdot b(l_{prev}-k)}$
18: $\quad\quad k = k + 1$
19: $\quad$ **while** $n_{l_{prev}-k} < 1$ **and** $k < l_{prev}$ ▷ Modification
20: $\quad l_{next} = l_{prev} - (k - 1)$ ▷ Modification
21: **end if**

A.2 Rate Adaptation Algorithm Validation

The authors [63] made two tests in order to prove that the QAAD works as described. In order to validate the implementation of the algorithm and the modifications made to it, the algorithm was tested in the same way and the obtained results were compared to the ones presented in the paper.

A.2.1 Step-down Test

In the step-down test, the available bandwidth is reduced from 2200 kbps to 800 kbps at 30 s. The following parameters are used: $\tau = 2$, $\mu = 10$, $\sigma = 3$, and the available bitrates are: $\{$400, 500, 600, 800, 1000, 1200, 1500, 2000$\}$ kbps. Figure A.1 presents the instantaneous throughput along the session.

Figure A.2 compares the results obtained in the paper with the results obtained with the implemented

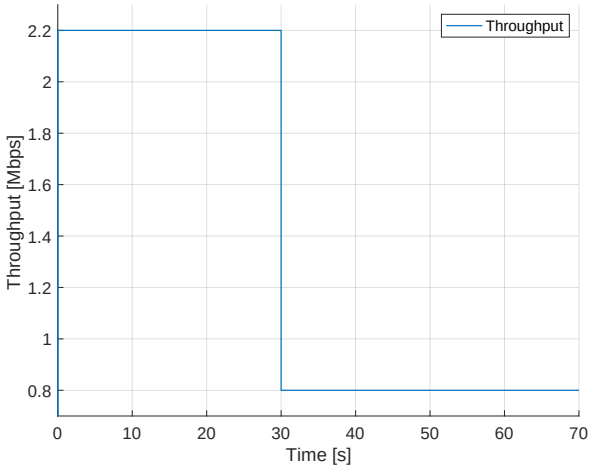
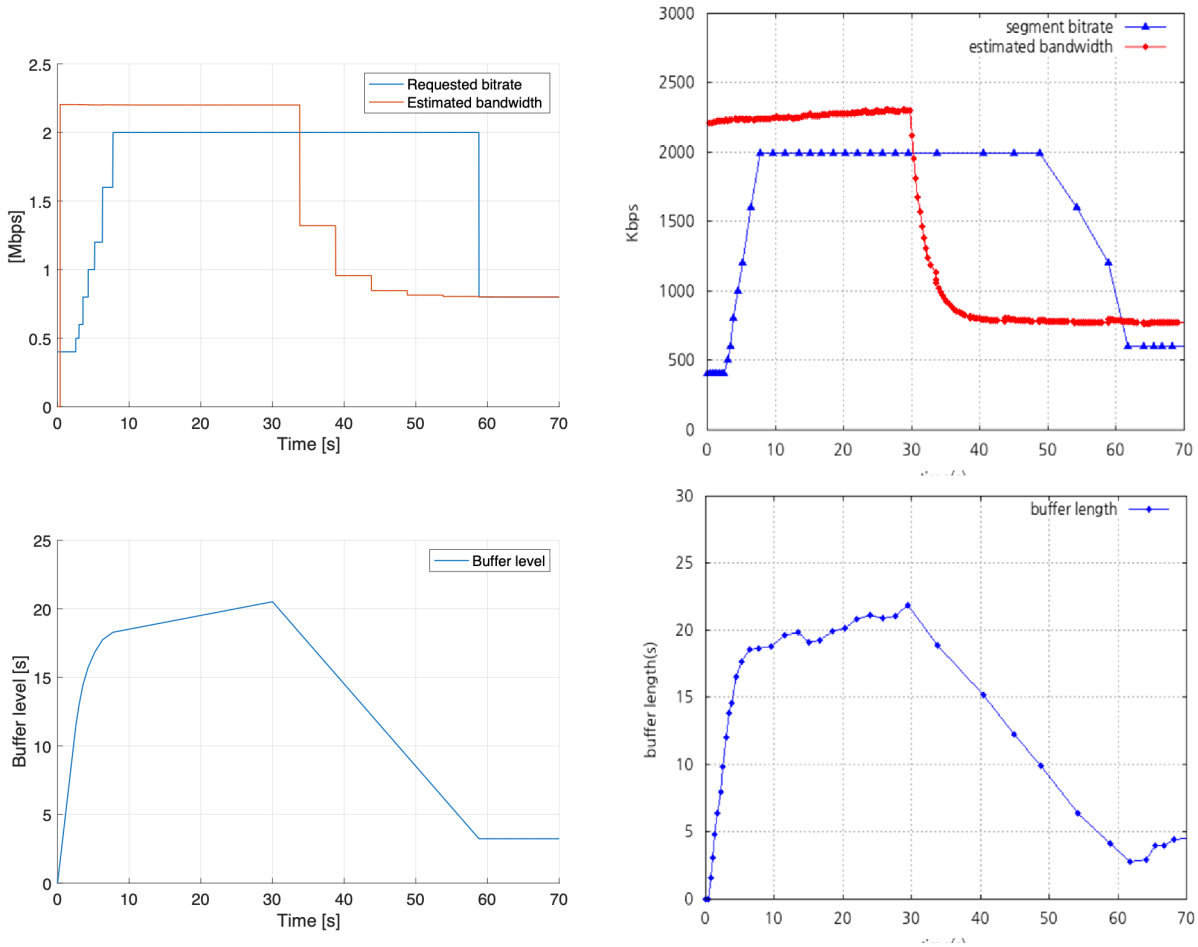


Figure A.1: Throughput experienced by the user in step-down test.



(a) Results obtained with the simulator.

(b) Results presented in [63].

Figure A.2: Comparison of the estimated bandwidth, requested segment bitrate and buffer level for the step-down test.

algorithm. It is possible to observe that the client starts by requesting the lowest quality level up until the marginal buffer level is reached. After this, since the available throughput is higher than the currently requested bitrate, the quality level starts to be incremented one by one, until it reaches the maximum

quality, as the bitrate for this level is still below the available throughput. When the available throughput drops, the estimated throughput starts to drop as well, smoothed by the moving average. It is possible to observe that the implemented throughput estimation scheme, although slower and with less updates than the one originally used one, still manages to have a good enough accuracy. The client notices that the available bandwidth becomes lower than the requested bitrate, but tries to hold the quality until the buffer reaches the minimal level. When this happens, the client selects the maximum feasible quality level in order not to let the buffer completely deplete. It happens that there is an available bitrate with the same value of the throughput, and as such the buffer level remains stable until the end of the simulation.

It is also possible to observe that the simulator performs the quality decrease at once instead of in several steps as in the paper, and that the final quality level is higher than the one in the paper. This probably has to do with the fact that the experiments in the paper were made with a real server and client, and therefore the recorded bandwidth has small fluctuations that the simulator does not contemplate.

A.2.2 Fluctuation Test

In the fluctuation test, the available bandwidth is fluctuated repeatedly every 4 seconds from 2200 kbps to 800 kbps and again from 800 kbps to 2200 kbps. The other parameters are the same as the step-down test. Figure A.3 presents the instantaneous throughput experienced by the user.

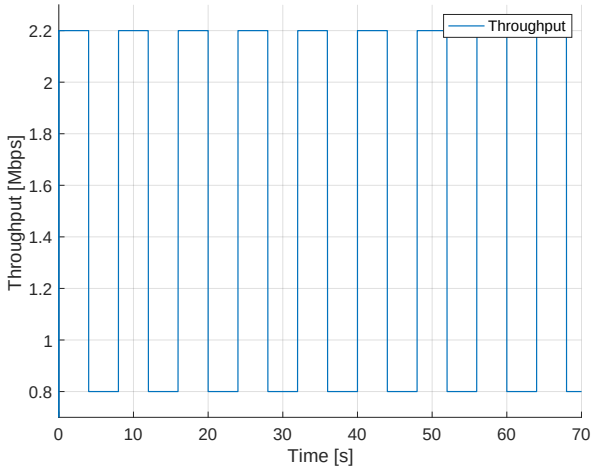
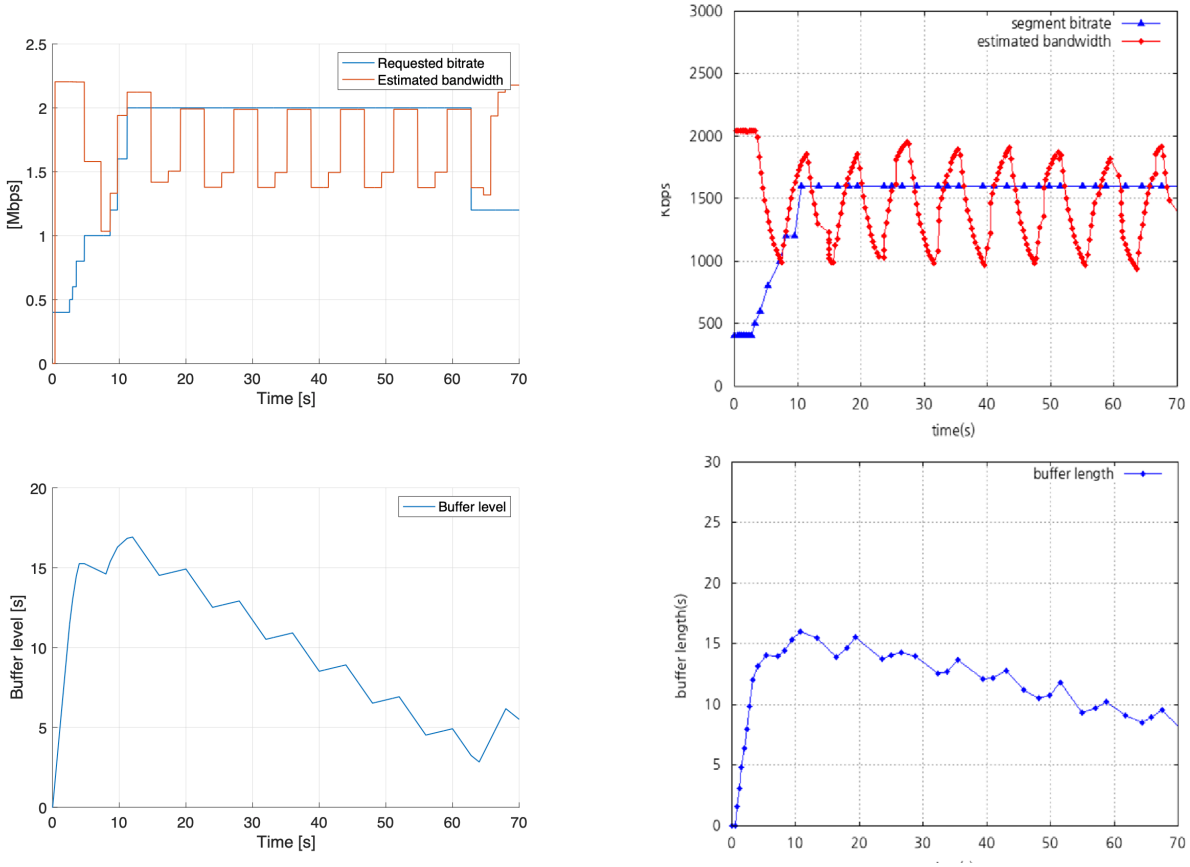


Figure A.3: Throughput experienced by the user in fluctuation test.

Figure A.4 compares the results obtained in the paper with the results obtained with the implemented algorithm. It is noticeable that using the segment-based bandwidth estimation scheme causes a worse adaptation behavior in this scenario. Because it is not fast enough to accurately track the bandwidth fluctuations, the client requests a quality level with a bitrate that is too high in relation to the available bandwidth, and the buffer depletes much faster than in the paper. This causes the client to reach the minimal buffer level and forces it to perform a quality drop in order to avoid stalling. However, the fluctuation test is a worst-case scenario, as in reality the available bandwidth varies in a much smoother way over time, giving the estimation algorithm more time to produce an accurate estimate.



(a) Results obtained with the simulator.

(b) Results presented in [63].

Figure A.4: Comparison of the estimated bandwidth, requested segment bitrate and buffer level for the fluctuation test.

Appendix B

PF Algorithm

The pseudocode of the implementation of the PF algorithm is presented in Algorithm 4.

Algorithm 4 Pseudocode of the PF algorithm.

get array of valid users (users that are requesting data)
for each valid user i **do**
    $m_{i,k} = \frac{r_k^i(t)}{r^{(avg)}}$
end for
for each RB k **do**
    get the highest metric
    allocate k-th PRB to user i with the highest metric
    $\delta_{ik} = 1$
    recalculate the metric for user i according to $m_{i,k} = \frac{r_k^i(t)}{r^{(avg)} + \sum_{k=1}^{K} \delta_{ik} r_k^i(t)}$
    if PRBs allocated to user i are enough to fulfill its request **then**
        remove user i from array of valid users
    end if
end for

B.1 Scheduling Algorithm Validation

In order to validate the implementation of the algorithm, an experiment was made where two users are streaming for 1 second. User 1 has a constant CQI of 10 and User 2 has a constant CQI of 15, as shown in Figure B.1. As such, the throughput achievable by User 2 is always greater than the one by User 1.

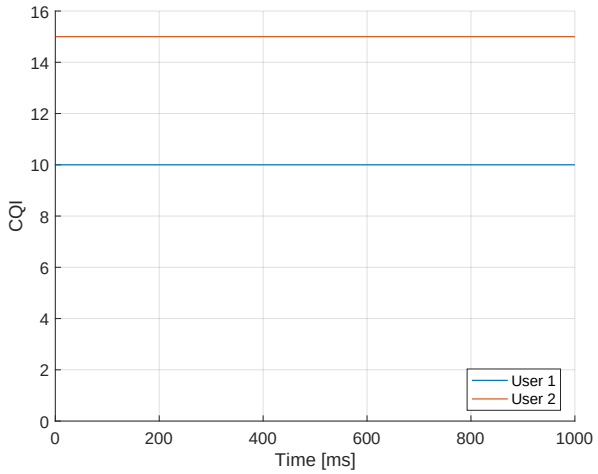
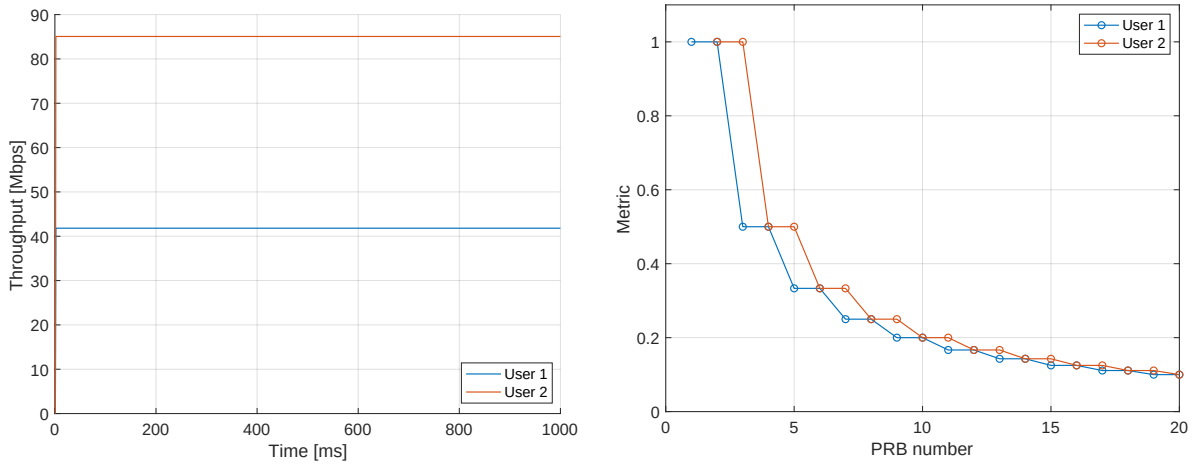


Figure B.1: CQI values reported by the users along the session.

The throughput experienced by the users is presented in Figure B.2(a). It is possible to observe that their throughputs are constant because the PF algorithm allocates both users the same number of PRBs every TTI. In fact, the algorithm equally distributes the available PRBs across both users even though one of them has a lower CQI. In order to gain more insight into what happens, Figure B.2(b) shows the metric evolution in the first TTI, for the first 20 PRBs.



(a) User throughput along the session.

(b) User metrics along one TTI.

Figure B.2: Simulated parameters.

Both users start with an infinite metric since $r^{(avg)}$ is initially zero. Therefore, the algorithm selects User 1 to receive the first PRB, which lowers their metric to 1. On the second PRB, User 2's metric is still infinite, so they are allocated the second PRB, lowering their metric to 1. On the third PRB, both users' metric is 1, so once again, the algorithm selects User 1 to receive the PRB, lowering their metric to 0.5. This keeps happening successively until there are no more PRBs to allocate.

Appendix C

Simulator Validation

In order to validate the behaviour of the simulator, a simulation was made while recording various user metrics throughout the simulation time. The simulation time is 60 seconds, using a tiles-based delivery scheme with 10 users. The results can be seen in Figure C.1.

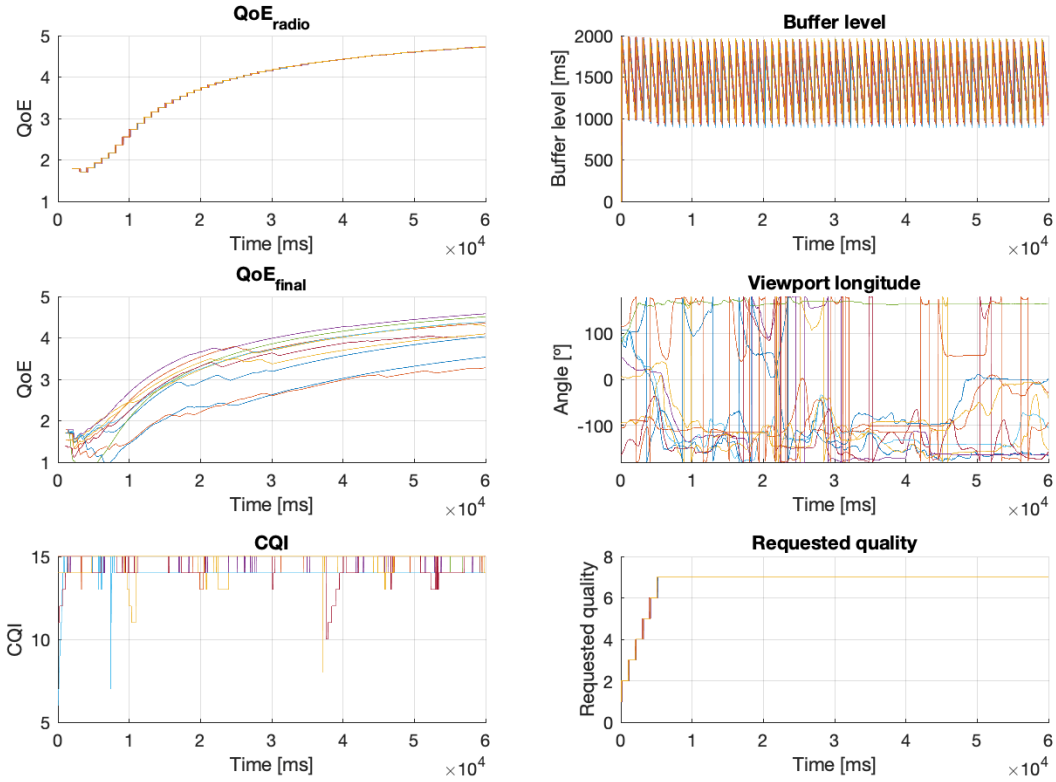


Figure C.1: User metrics throughout simulation.

Although there are some fluctutations in the CQI of the users, all users can be satisfactorily served since there are few. It is possible to observe that the requested quality is increased by one step each time a new segment is requested, which in this case happens each time the buffer falls below the threshold of 1000 ms. The QoE_{radio} and QoE_{final} metrics are further explained in Section 5.2. It is possible to see the QoE_{radio} continually increases until all users almost reach the maximum QoE by the end of the simulation. Because of the viewport trajectory, which is faster for some users than for others, some users will have a bigger penalization in their QoE_{final} than others.

Appendix D

Monte Carlo Method

As described in Section 5.1, some of the simulator parameters, such as the test sequence, CQI profile and viewport trajectory allocations, are stochastic processes. Naturally, this results in a random component of the input variables, which introduces a random component in the simulation results. Therefore, it is not sufficient to run the simulation once, as the accuracy of that result would be unknown. In order to obtain statistically relevant results, the Monte Carlo method is used. Each simulation is run a certain number of times with different random test sequence, CQI profile and viewport trajectory allocations each time, in order to derive a final result that includes a mean value and a confidence interval.

The final result is derived based on the Central Limit Theorem [76]. This theorem states that, if there is a random variable $X = \{x_1, x_2, x_3, \ldots, x_n\}$ with mean μ_x and variance σ_x^2, the arithmetic mean of the samples,

$$m = \frac{\sum_{i=1}^{n} x_i}{n}, \tag{D.1}$$

approaches a normal distribution with the random variable's mean, μ_x, and standard deviation $\sigma_x/\sqrt{n}$ as n approaches infinity, and it does so irregardless of the actual distribution of the random variable.

The samples' arithmetic mean is therefore the most probable solution, while the standard deviation defines the confidence interval. The confidence interval is an estimate that provides a range of values for a given parameter with a certain degree of confidence that the true value is within that range. It requires the specification of a confidence level, which is usually a large percentage, such as 95%. In other words, the 95% confidence level is a range of values in which it is possible to be 95% certain it contains the true mean of the parameter.

Based on this, it is possible to compute the appropriate number of Monte Carlo simulations that should be conducted in order to achieve a confidence interval of a particular width given a desired confidence level [77]. The minimum number of simulations, n, that should be run in order to achieve a certain target accuracy is given by

$$n = \left(\frac{z_{\alpha/2}}{w} \frac{\sigma_x}{\mu_x} \right)^2, \tag{D.2}$$

where $z_{\alpha/2}$ is the value of the normal probability distribution function for the half distance $\alpha/2$ and w is

the size of the confidence interval, normalized to the mean.

In this study, a maximum confidence interval of 1% of the mean ($w = 0.01$), with a probability of 95% (corresponding to $z_{\alpha/2} = 1.96$, as illustrated in Figure D.1) are assumed. This means that, if the simulations are run n times, the real solution is in the interval $[\hat{x} - w\hat{x}, \hat{x} + w\hat{x}]$ with 95% certainty.

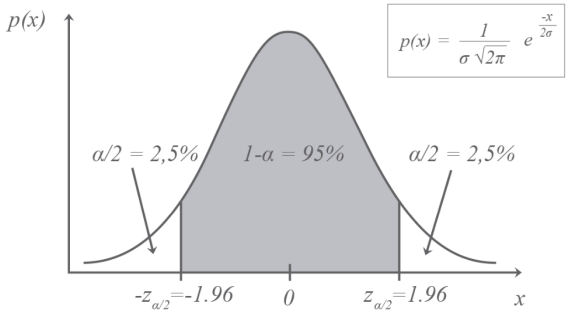


Figure D.1: Illustration of 95% confidence interval in the normal distribution [78].

Since it is not possible to know μ_x and σ_x a priori, it is necessary to first run an arbitrary large number of Monte Carlo simulations. Then, the resulting μ_x and σ_x values are calculated and used to compute the minimum number of simulations, n, according to (D.2). If the resulting n is lower than the already run number of simulations, the confidence interval requirements are met and no further simulations are needed. If n is larger than the number of run simulations, more simulations need to be run.

D.1 Monte Carlo Method Demonstration

In order to demonstrate the Monte Carlo method, 12 Monte Carlo simulation were run using the simulation parameters described in Section 5.1, and using the Monolithic scheme with a latency of $L = 10$ ms and a buffer threshold of $\alpha = 6000$ ms. Each simulation was ran for 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 users. From each number of users to the next, the random allocations are kept. In other words, for example, in the simulation for 20 users, the first 10 users have the same allocations as the users in the 10 users simulation. Figure D.2 plots the percentage of connected users that have a $QoE_{final} \geq 3$ as a function of the number of connected users.

The Monte Carlo curve is an average of the values obtained in each simulation. Performing a significant number of simulations increases the accuracy of the results, guaranteeing with 95% certainty that the real solution does not differ by more than 1% from the presented values.

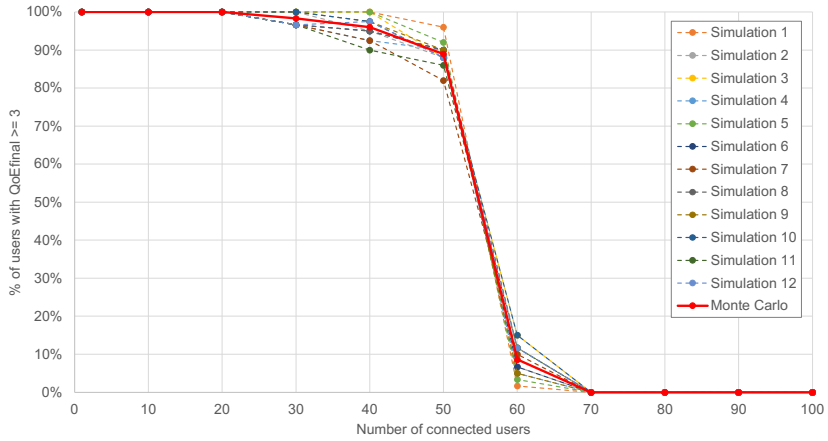
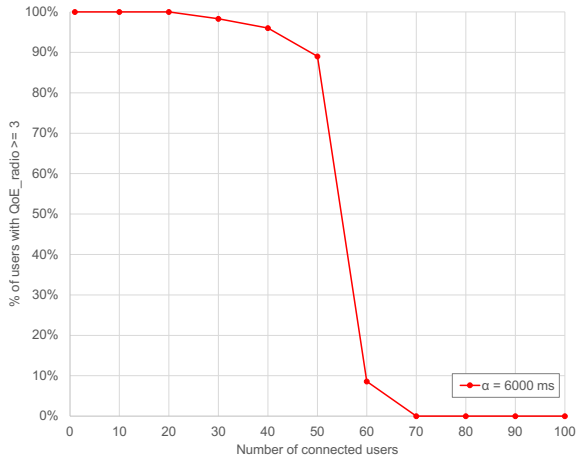


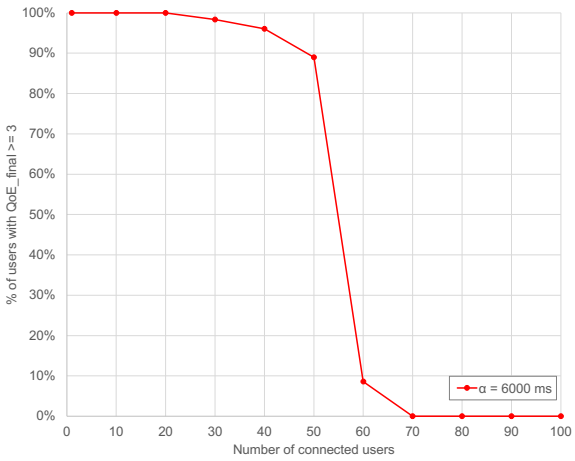
Figure D.2: Monte Carlo method demonstration. Monolithic scheme, $L = 10$ ms, $\alpha = 6000$ ms.

Appendix E

Graphs of the Sensitivity Analysis

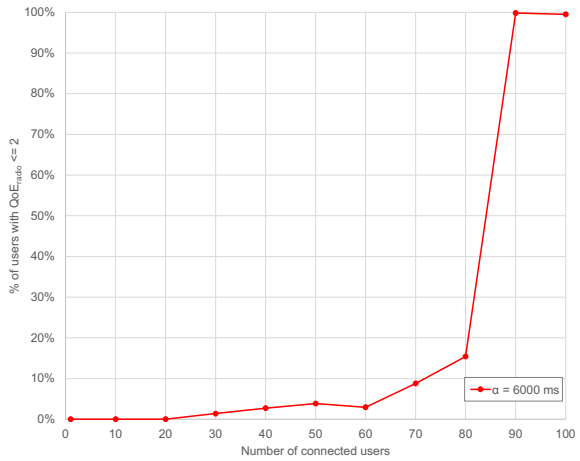


(a) QoE_{radio}

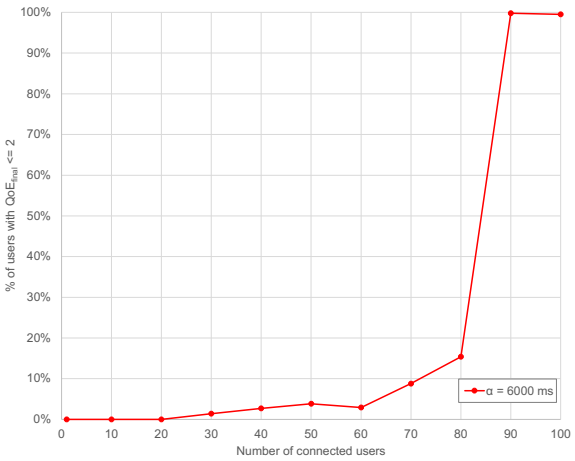


(b) QoE_{final}

Figure E.1: Percentage of satisfied users (S = 3). MonoEqui scheme, L = 10 ms.

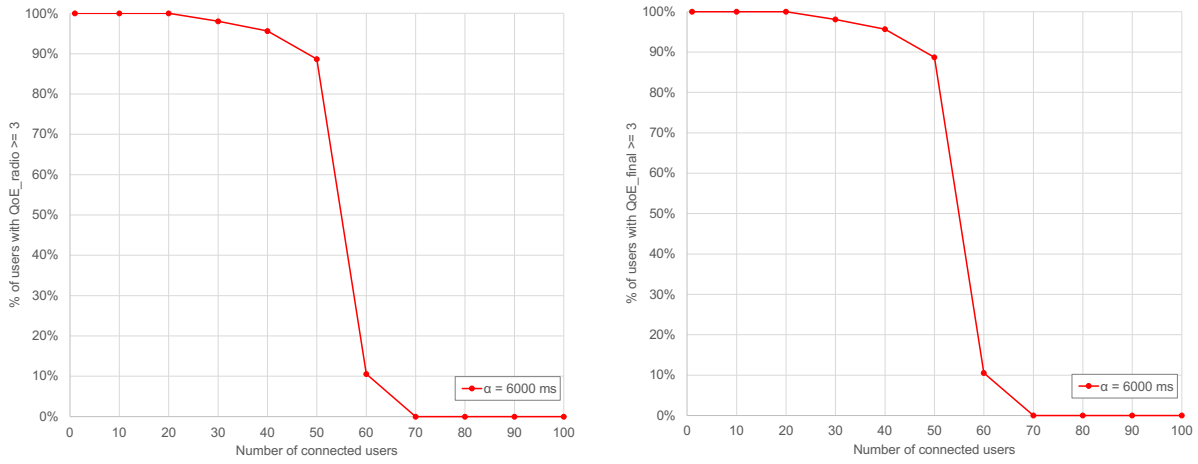


(a) QoE_{radio}



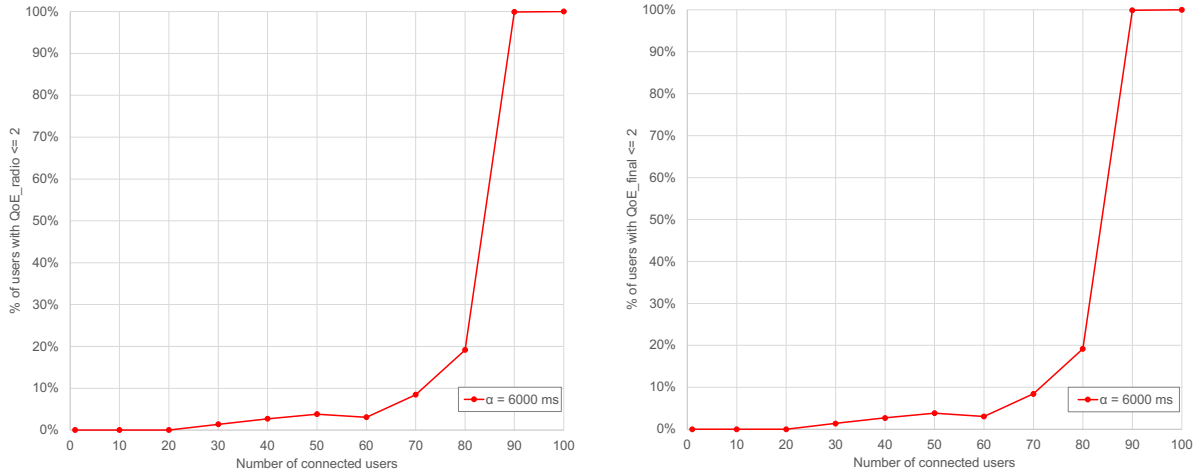
(b) QoE_{final}

Figure E.2: Percentage of non-satisfied users (NS = 2). MonoEqui scheme, L = 10 ms.



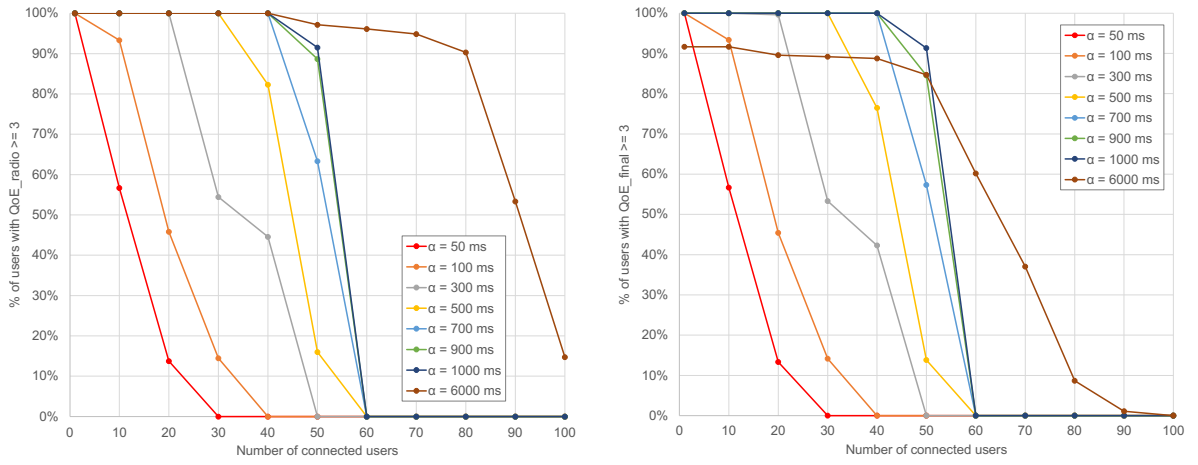
(a) QoE_{radio} (b) QoE_{final}

Figure E.3: Percentage of satisfied users (S = 3). MonoEqui scheme, L = 1 ms.



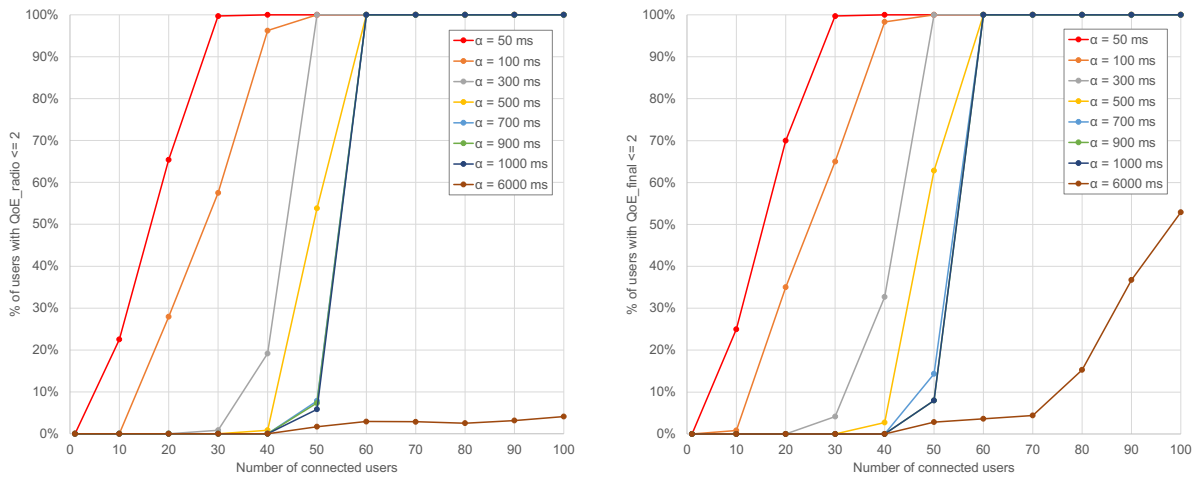
(a) QoE_{radio} (b) QoE_{final}

Figure E.4: Percentage of non-satisfied users (NS = 2). MonoEqui scheme, L = 1 ms.



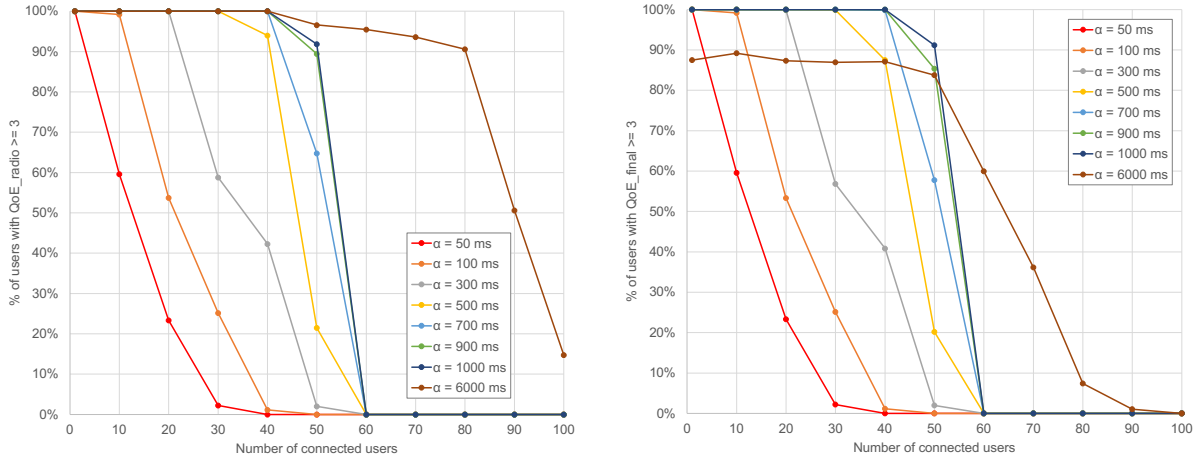
(a) QoE_{radio} (b) QoE_{final}

Figure E.5: Percentage of satisfied users (S = 3). OMAF-SRes scheme, L = 10 ms.



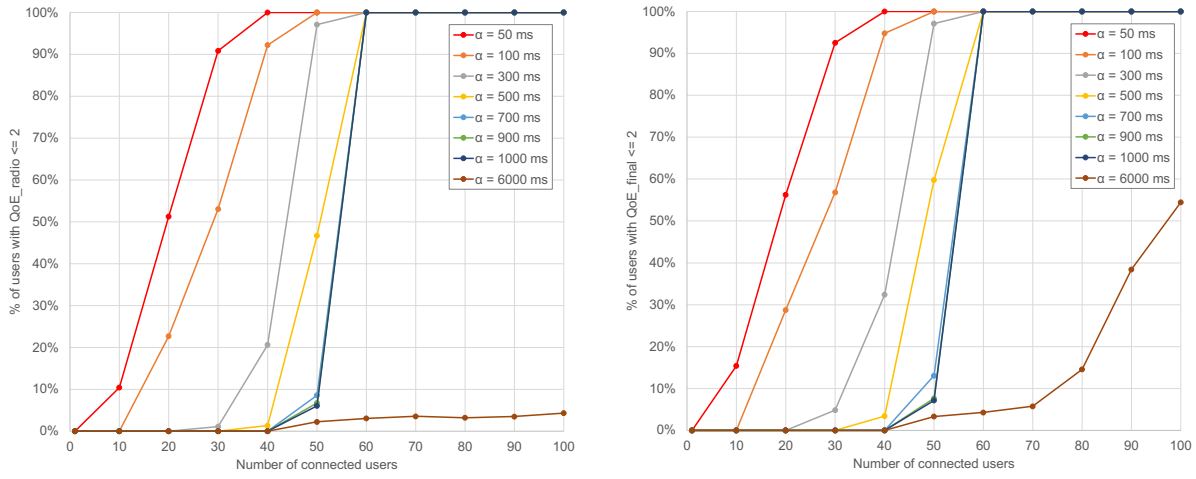
(a) QoE_{radio} (b) QoE_{final}

Figure E.6: Percentage of non-satisfied users (NS = 2). OMAF-SRes scheme, L = 10 ms.



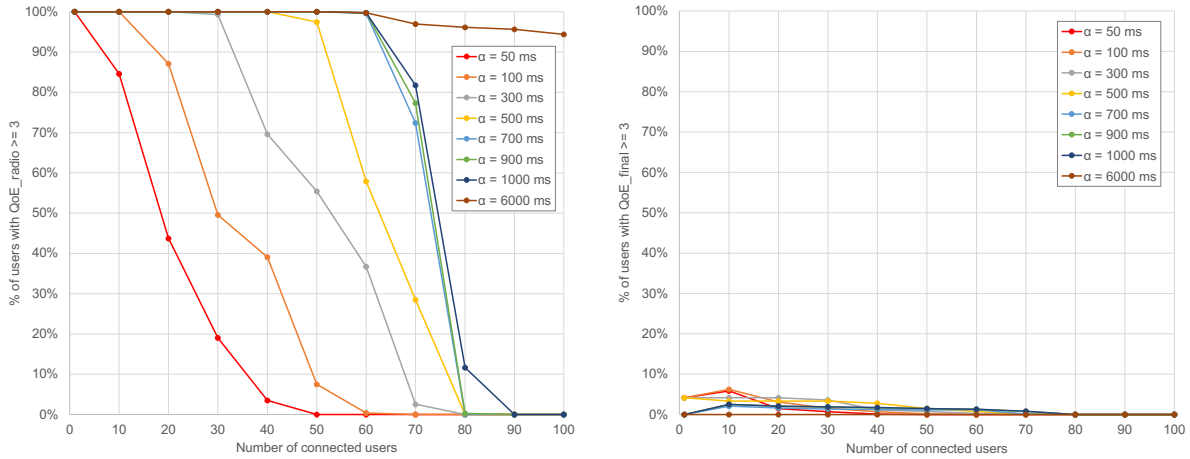
(a) QoE_{radio} (b) QoE_{final}

Figure E.7: Percentage of satisfied users (S = 3). OMAF-SRes scheme, L = 1 ms.



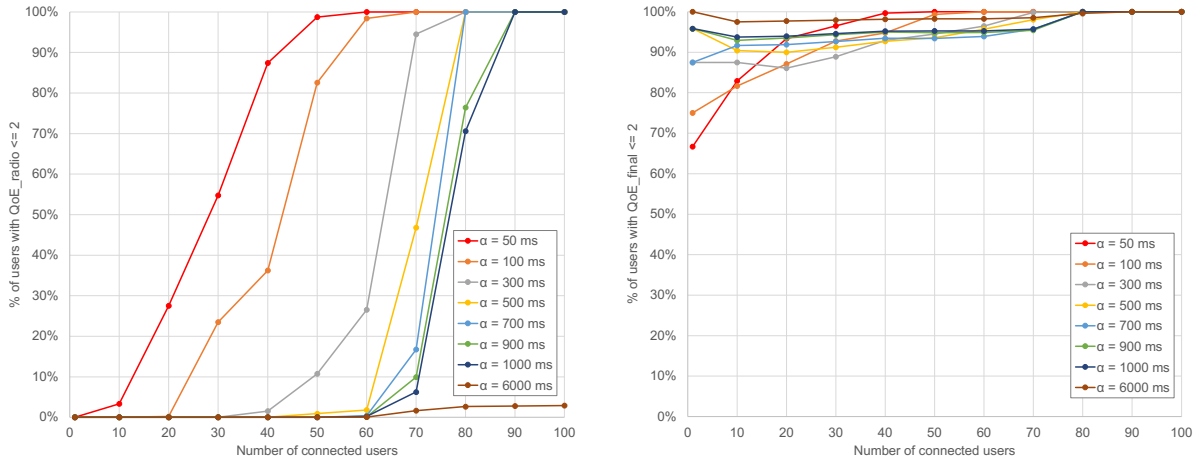
(a) QoE_{radio} (b) QoE_{final}

Figure E.8: Percentage of non-satisfied users (NS = 2). OMAF-SRes scheme, L = 1 ms.



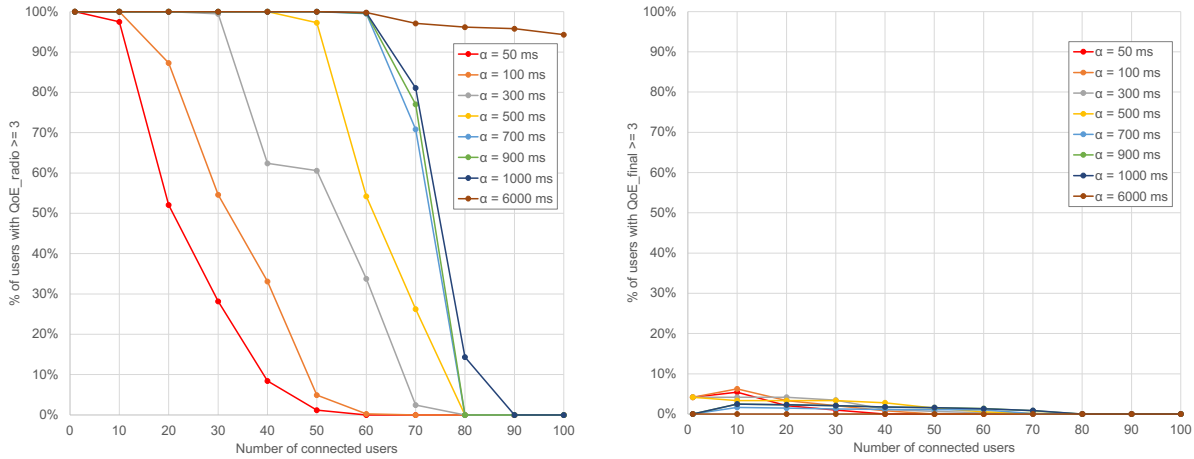
(a) QoE_{radio} (b) QoE_{final}

Figure E.9: Percentage of satisfied users (S = 3). OMAF-SRes-Partial scheme, L = 10 ms.



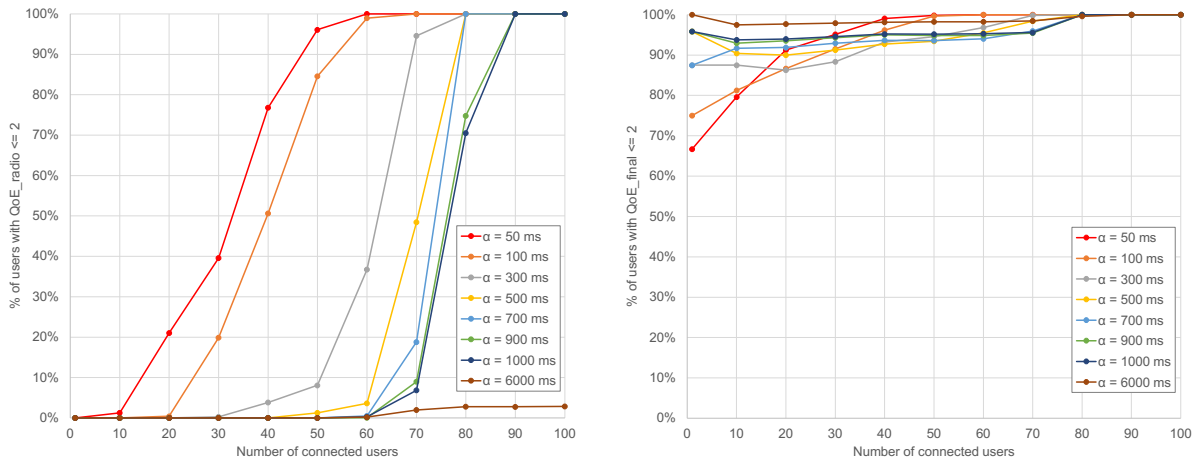
(a) QoE_{radio} (b) QoE_{final}

Figure E.10: Percentage of non-satisfied users (NS = 2). OMAF-SRes-Partial scheme, L = 10 ms.



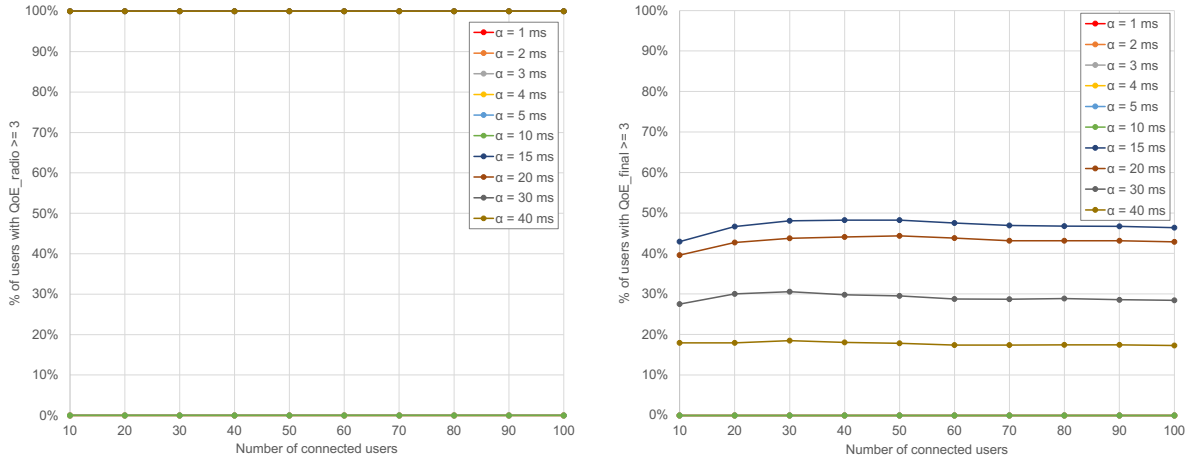
(a) QoE_{radio} (b) QoE_{final}

Figure E.11: Percentage of satisfied users (S = 3). OMAF-SRes-Partial scheme, L = 1 ms.



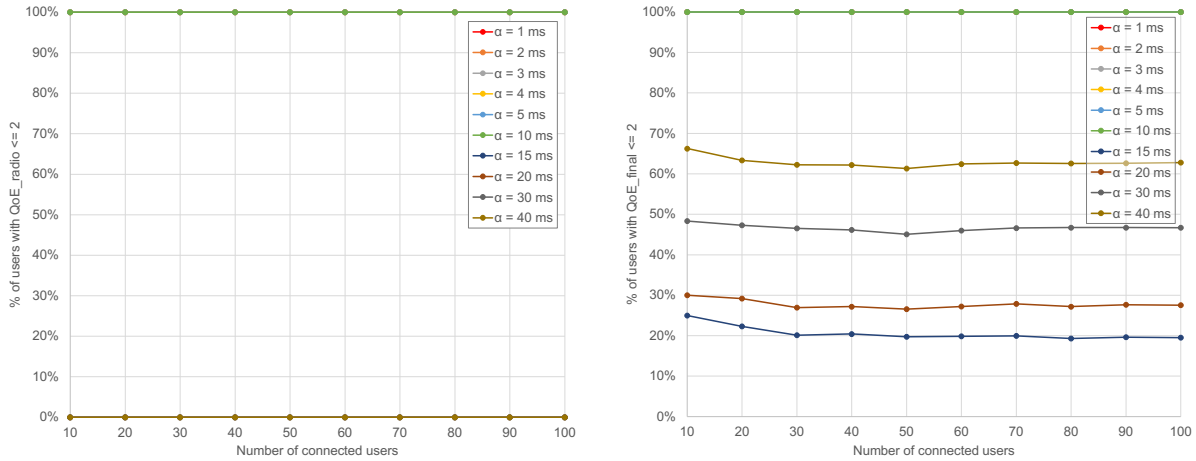
(a) QoE_{radio} (b) QoE_{final}

Figure E.12: Percentage of non-satisfied users (NS = 2). OMAF-SRes-Partial scheme, L = 1 ms.



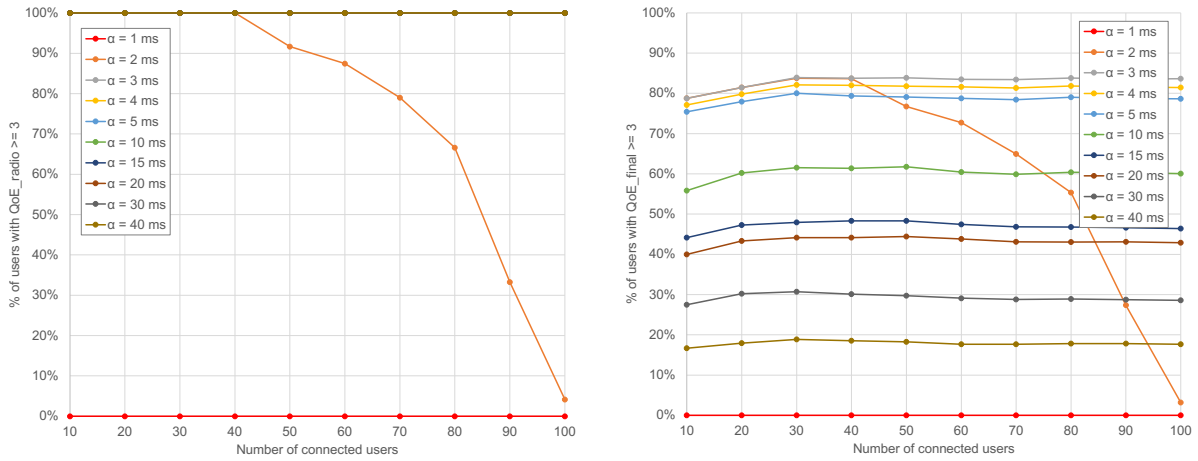
(a) QoE_{radio} (b) QoE_{final}

Figure E.13: Percentage of satisfied users (S = 3). Viewport-Only scheme, L = 10 ms.



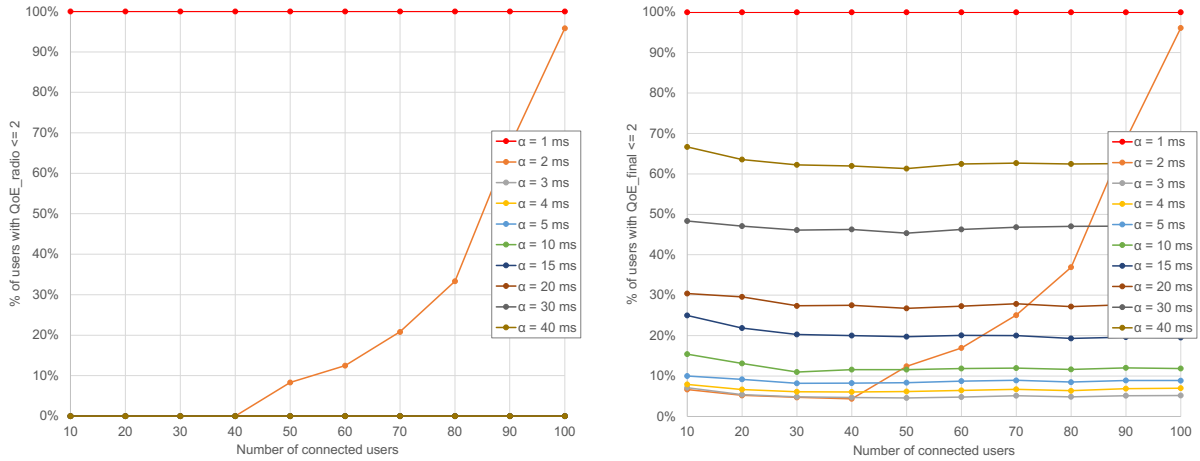
(a) QoE_{radio} (b) QoE_{final}

Figure E.14: Percentage of non-satisfied users (NS = 2). Viewport-Only scheme, L = 10 ms.



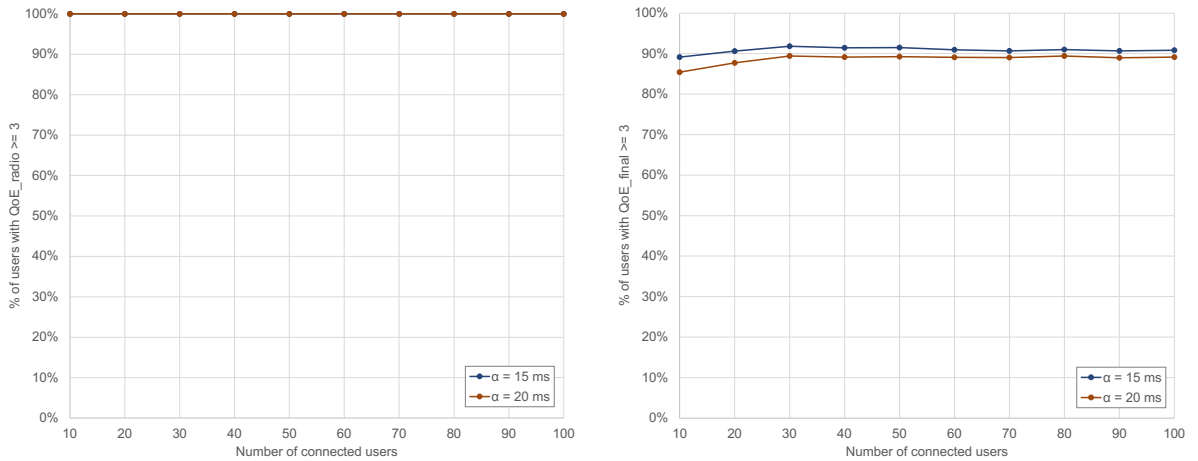
(a) QoE_{radio} (b) QoE_{final}

Figure E.15: Percentage of satisfied users (S = 3). Viewport-Only scheme, L = 1 ms.



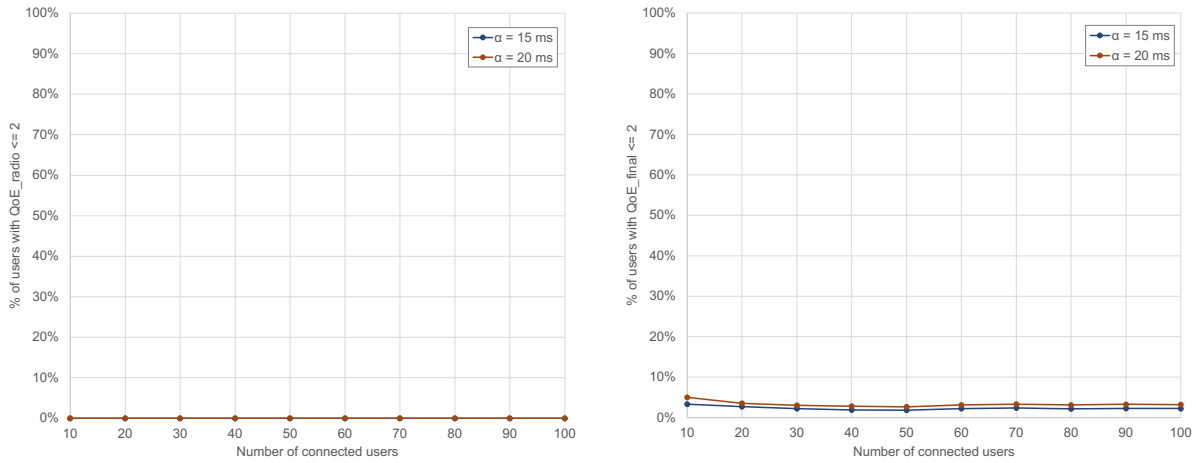
(a) QoE_{radio} (b) QoE_{final}

Figure E.16: Percentage of non-satisfied users (NS = 2). Viewport-Only scheme, L = 1 ms.



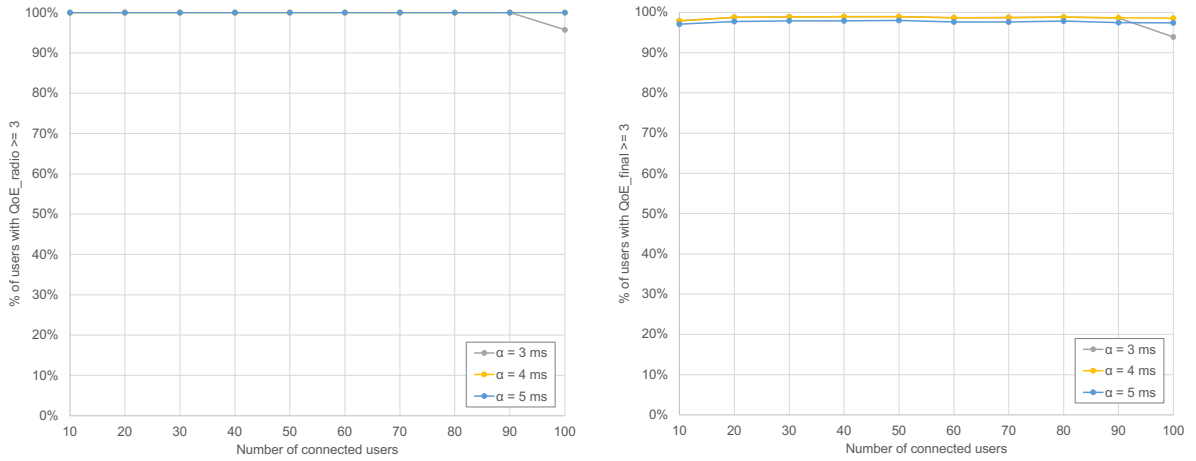
(a) QoE_{radio} (b) QoE_{final}

Figure E.17: Percentage of satisfied users (S = 3). Viewport-Only-Margin scheme, L = 10 ms.



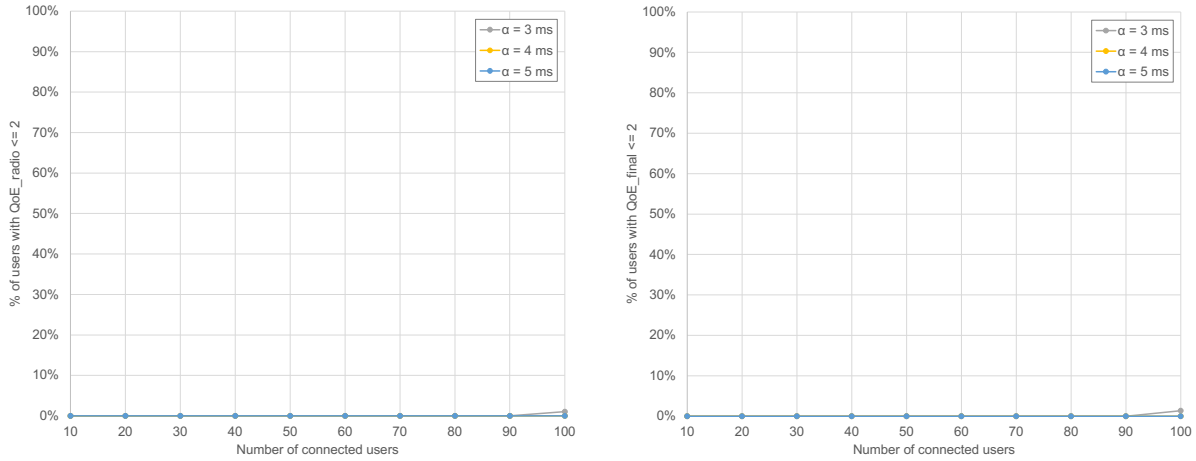
(a) QoE_{radio} (b) QoE_{final}

Figure E.18: Percentage of non-satisfied users (NS = 2). Viewport-Only-Margin scheme, L = 10 ms.



(a) QoE_{radio} (b) QoE_{final}

Figure E.19: Percentage of satisfied users (S = 3). Viewport-Only-Margin scheme, L = 1 ms.



(a) QoE_{radio} (b) QoE_{final}

Figure E.20: Percentage of non-satisfied users (NS = 2). Viewport-Only-Margin scheme, L = 1 ms.