FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

New interaction models for 360º video

Gonçalo Teixeira da Costa

FOR JURY EVALUATION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Maria Teresa Andrade (PhD)

Co-Supervisor: Tiago Soares da Costa (MSc)

June 23, 2019

Resumo

Esta dissertação pretendeu estudar mecanismos e abordagens para optimizar a qualidade de experiência dos utilizadores de um sistema de streaming de conteúdos multivista e, consequentemente, conceber e testar uma solução adequada para alcançar esse objectivo. Deste modo, tendo presente a minimização dos recursos computacionais e de rede utilizados.

Como ponto de partida para este desafio foi utilizado um protótipo laboratorial de um sistema de streaming multivista, anteriormente desenvolvido pelo INESC TEC. Este Protótipo foi desenvolvido no âmbito de um projecto de investigação e não oferecia os níveis de desempenho desejados. O protótipo existente recorre à norma MPEG-DASH para permitir comutar entre diferentes fluxos de dados ao longo da transmissão e assim adaptar o contéudo de acordo com as exigências do contexto de utilização, tais como restrições de largura de banda, ou de acordo com as preferências do utilizador, tal como a área da imagem em que o utilizador está mais interessado, isto é, o foco de atenção do utilizador na cena. Com este propósito, faz também uso de um sistema que permite detectar o foco de atenção do utilizador, o qual se baseia na análise de imagens recolhidas por uma Webcam apontada para o utilizador.

A versão inicial do sistema não tem os resultados desejados e, como tal foi proposto a análise do problema e a apresentação de uma abordagem capaz de ultrapassar limitações existentes, nomeadamente a latência significativa no cliente quando o utilizador muda o seu foco de atenção. Deste modo, esta dissertação pretende identificar as dificuldades que se colocam relativamente à disponibilização e transmissão eficiente deste tipo de conteúdos, assim como os compromissos necessários ao nível da qualidade de experiência do utilizador.

A abordagem que foi concebida passa pela incorporação de um mecanismo de buffering no sistema. O mecanismo desenvolvido é composto por um proxy e tem como objectivo de minimizar o atraso existente na transição de vistas. O proxy será capaz de identificar os pedidos que deve fazer so servidor em nome do cliente e gerir os diferentes pacotes que recebe, enviando apenas um ao cliente. A incorporação deste mecanismo vem propocionar melhorias na qualidade de serviço e na qualidade de experiência

Abstract

Today, the fast technological evolution and the significant increase in the demand for multimedia content has boosted the development of the transmission mechanisms used for this purpose. This development had repercussions in several areas, such as immersive experiences that include 360º contents. Whether through live streaming or by using on demand services, the quality of service and experience have become two points whose development has assumed high importance.

The capture and reproduction of 360º content allows transmission of an immersive view of reality at a given moment. With this approach, the industry intends to provide a product with better audiovisual quality, more comfortable for the user and which allows better interaction with the same. An example of this regards the the selection of viewing angles that most appeals to us in a given event (for example, football matches or concerts).

This dissertation has as main objective the incorporation of a buffering mechanism in a multimedia system, able to offer adaptive multi-view experiments. The system uses the MPEG-DASH protocol for efficient use of network resources and a conventional camera for detecting the movements of the user's head and selecting the points of view that one wishes to visualise in real time. The system also incorporates an automatic quality adjustment mechanism, adjustable to the network conditions.

The buffering mechanism is intended to increase the quality of experience and the quality of service, minimising the delay in the transition between views. The mechanism will consist of a proxy capable of sending three views simultaneously. Of these views, two will be sent in low quality, while the main view will be sent and presented to the user in high quality. Whenever there is a new request from the user, the mechanism will switch between sent views until it receives the response from the server.

Based on these assumptions, the dissertation intends to identify the challenges that are posed regarding the availability and efficient transmission of 360º content, as well as the necessary commitments regarding the quality of user experience. This last point is particularly significant, taking into account the network requirements and the volume of data presented by the transmissions of this type of content.

Agradecimentos

Agradeço à minha orientadora, a Professora Doutora Maria Teresa Andrade, pela orientação e por ter desafiado os meus conhecimentos académicos com esta proposta de trabalho.

Ao meu co-orientador, o Engenheiro Tiago Soares da Costa, pelo apoio prestado ao longo da realização desta dissertação, pela acessibilidade, diponibilidade e prontidão.

Agradeço à Faculdade de Engenharia da Universadede do Porto por me providenciar uma formação académica diversificada e de qualidade, passando sempre os melhores valores de trabalho.

Ao INESC TEC pelo espaço e condições de trabalho cedidos para o desenvolvimento desta dissertação.

Quero agradecer à minha mãe e ao meu pai por me terem proporcionado as melhores condições e pelo apoio incondicional ao longo da minha formação académica.

À minha irmã pela prontidão, pelos conselhos e apoio incondicional dados ao longo dos anos.

Quero agradecer à Maria Inês por estar presente em todos os momentos e pelo apoio que sempre disponibilizou.

Ao Fraga pelo companheirismo, apoio e amizade, e pelas longas noites de trabalho na faculdade.

Por fim, agradeço à minha família e aos meus amigos pelo apoio dado e por contribuirem indirectamente para a realização desta dissertação e para a minha formação académica .

Gonçalo Costa

"Always do your best.
What you plant now, you will harvest later."

Og Mandino

Contents

List of Figures

List of Tables

Symbols and Abbreviations

2D	Two-Dimensional
3DOF	Three Degree of Freedom
3D	Three-Dimensional
6DOF	Six Degree of Freedom
AI	Artificial Intelligence
AOMedia	Alliance for Open Media
ARP	Address Resolution Protocol
AV1	AOMedia Video 1
AVC	Advanced Video Coding
CPU	Central Processing Unit
DASH	Dynamic Adaptive over HTTP
DOF	Degree of Freedom
FIFO	First In First Out
FTP	File Transfer Protocol
HEVC	High Efficiency Video Coding
IP	Internet Protocol
MFU	Media Fragment Unit
MMT	MPEG Media Transport
MPD	Media Presentation Description
OS	Operating System
QoE	Quality of Experience
QoS	Quality of Service
RTCP	Real Time Transport Control Protocol
RTP	Real Time Transport Protocol
SRD	Spatial Relationship Description
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URI	Uniform Resource Identifier

Chapter 1

Introduction

This chapter aims to introduce the theme. It provides information on the context of the problem and an overview over the dissertation theme. It is also composed by the motivation, the description of the problem studied, the structure of the document and the achieved work.

1.1 Motivation

At present we can experience reality virtually using the TV broadcast. Thus comes the comfort of home entertainment that gives us the feeling of immersion in content. [10] [11]

The need to support the development of communication technologies and devices, tends to increase as their usage has been increasing. The search for content that provides immersive experiences has increased, and due to this fact technologies have been developed to provide this type of experience in live events such as concerts or sporting events. [12]

The 360º experience represents a good solution to provide a virtual perspective of reality. The industry's main objective is to offer a product with the best visual quality, in order to provide to the user a comfortable and better interaction. The experience with 360º content has the advantage of providing access to content that may be restricted to live performances or offer a lower cost experience, but will never fully reproduce the full sensory experience, being only an approximation of the live event and will always take into account this disadvantage. [12]

Consumers want more immersive experience, and with the increase of their demand for this type of content some companies seek to develop facilities for content access. Since 2006, the social network Facebook invests on 360º content, offering experience with 360º video in 2017 it improved this experience allowing to see live performances in 360º and 4K quality. These improvements come with the aim of offering better and more immersive video streaming experiences. With these advances in the provision of multimedia content, is expected an increase in the list of compatible equipment and programs. [13]

Immersion brings a new era for entertainment experiences. We can experience everyday situations in a realistic way. Three factors are necessary to approximate reality: visual quality, sound

quality and interactivity. [14] There are different equipment, technologies and hardware that provide immersive experiences. The disadvantage of this kind of equipment is the cost. However, there are more affordable consumer solutions, with lower quality. [15]

Immersive technologies have been mostly sought after by the public for their ability to create experiences which are closer to reality. These kinds of experiences are interacting with the intimate audience creating a powerful memory. [12] These experiences cover all types of multimedia entertainment, from movies, video games and music. The demand for consumer entertainment now points to a new phase where immersive experiences are the most important factor. [15]

1.2 Problem Description

This dissertation is proposed as a result of the research project carried out by INESC TEC's Multimedia and Telecommunications Center under the name of "Multiview Personalized Experience", which aims to create a complete FTV system in a client-server architecture. The system is able to provide an immersive experience by providing the user with a system which adapts to the bandwidth provided, and allows the user to control the 3D video viewpoint.

The prototype was built on the basis of MPEG-DASH and includes the following services:

- HTTP Server: Contains the media content of multiple views of a single scene capturing with 2D cameras positioned around the scene and distanced by a specific angle;
- Media Player: Responsible for the transmission and reproduction of the content, and manages all operations requested by the client;
- Client application: Interprets the information of head tracking and sends this information and other parameters to the media player. Information can also be sent by manual control;
- Webcam: Responsible for head tracking data, extracted from users.

The prototype has a dynamic quality mechanism, in order to be able to adapt to different network throughput levels. During the experiment, you may request a different view. The prototype has no buffering for view switching purposes, introducing a latency problem during view transitions. Therefore, the main concern of this dissertation is to reduce this latency through the implementation of a buffering mechanism applied to view switching operations, taking into account QoE and QoS issues.

The main objective of this dissertation is to specify and develop a multi-view agent/gateway for IP-based multi-view streaming applications. Multi-view applications are capable of changing the perspective of the presented scene according to the viewer's attention focus, and doing this with minimal latency and without compromising the quality of the viewer's experience.

Based on the previous known 360^{o} multi-view streaming system, which uses the MPEG-DASH standard for encoding content and head tracking information according to the user' s focus

of attention, it' s intended to develop a new buffering associated with this system, in order to optimise the multimedia resources sent through the network and to decrease the latency each time the user changes its focus of attention.

In order to achieve this objective, it is necessary to apply buffering mechanism in C and determine this way which mechanisms are available for the MPEG-DASH and TCP/IP standard. When there is no available mechanism for this purpose, we offer new ones so that the system is able to achieve its set goals. Comparative tests for performance and latency were designed and performed before and after our improvements to the system in order to validate the viability of the offered solutions.

1.3 Document Structure

This dissertation is composed of five more chapters.

Chapter 2 describes the state of the art. Contextualising the theme of the dissertation and presenting related works.

In Chapter 3 the methodology is presented. Which qualitative parameters were studied and the description of the tests that were performed.

Chapter 4 presents an overview of the prototype and the incorporation of the proxy in the general architecture.

Chapter 5 presents the performance and efficiency tests and the results obtained.

Finally, in Chapter 6 we have the conclusions of the work accomplished and an approach to a future work.

1.4 Article

A brief article was written summarising the work carried out throughout the dissertation. It contains a description of the project, the objectives of the work and the conclusions of the work completed. This article was written in accordance with the requirements imposed by the faculty.

1.5 Website

A website was developed during the dissertation. Contains all important information about the dissertation. Composed of an introduction, motivation, objectives and team work. The site is available at `https://newinteraction271938301.wordpress.com/`.

Chapter 2

State of the Art

This chapter constitutes a review of the State of the Art. In this section, a description of the study that was performed is provided. This previous study contains approaches, techniques and solutions relevant to proceed with research and the work intended in this dissertation. This previous study allowed useful insights about relevant themes and a consolidation of knowledge, which served as basis for the developed work. This chapter approaches several concepts such as coding, data transmission and also other themes related to this dissertation, such as immersive experiences and three-dimensional space.

2.1 Video Coding and Compression

Perceptual coding was conceived in order to compress with minimum distortion to the observer, for that, basically, part of the information is eliminated reducing the amount of bits used. This way, redundant data, that can be obtained using other data in the signal, can be eliminated from the original content. On the other hand, irrelevant data, which can bring additional information for the signal is also not necessary to be preserved as it is not perceived by the observer.

There are essentially three types of redundancy: Spatial, temporal and statistical redundancy. To eliminate or reduce these redundancy some tools were designed, such as quantisation, prediction with motion estimation and compensation and entropy encoding, respectively. There is another type of redundancy that should be mentioned, spectral redundancy, that as to do with separating colour information from brightness using a collar systems. YCrCb model is used to represent colour space at video compression, which provides better perceptual efficiency.

There are several compression tools that can be used all together in diversified ways and, because of that, a number of standards for video compression were established to specify how to use these tools and the format of the final compressed bitstream. The conversion of a digital video to a format which takes less capacity when it is stored or transmitted. Video compression, is an essential technology for applications such as digital television, DVD-Video, mobile TV, video conferencing and Internet video transmission.

2.1.1 Advanced Video Coding

Video Coding Experts Group (VCEG) and the ISO/IEC MPEG developed a standard codec H.264/MPEG-4 Part 10 or H.264/AVC [16]. This codec uses an hybrid approach, because it results from the combined use of several compression tools along the video, and a toolbox approach, in which allows to select from a pool of compression tools and even different sub-sets of these tools different codecs in order to compress the video content and deliver the output of the codec a compatible bitstream. This entire section is based on [17].

Video content information is compressed on a block basis, while processed by the H.264/AVC/MPEG-4 Part 10, creating three modes of compression. Each of these three modes uses different subsets of the whole set of compression tools and also take in common a restricted number of tools, especially those that eliminate statistical redundancy. From this process results different types of compressed images as the following presented above:

- Intra mode: Delivering I frames. In this mode, when using spatial transforms, such as the Discrete Cosine transform, to blocks of image pixels and then uniform weighted quantisation, it is possible to reduce spatial redundancy and irrelevant information in the signal. Quantisation gives more importance to the low spatial frequencies of signal components that are those in which the human eye is more sensitive to detect distortion. When it is given more importance, in fact, it is being quantised at a higher level of quantisation;

- Predictive mode: Delivering P frames as result of a predictive compression with backwards motion-compensation. This mode predicts on a block basis, the movement that occurred from one frame to some previous image and represents that movement in a motion vector in a XY plane and outputs the difference between the real image and the prediction. Usually, this difference mentioned is very small and it will provide a smaller number of bits to encode. The tools used on this mode are the same as in the Intra mode;

- Bidirectional mode: Delivering B frames as a result of Predictive compression with forward and backwards motion compensation. It is similar to the Prediction mode predicting motion on a block basis, with the particular difference that this mode predicts the motion in relation to some previous and also to some future images and represents the motion in two motion vectors on a XY plane and also outputs the difference between the estimation and the real image.

H.264 has the necessary efficiency and flexibility to respond to different application requirements and because of that is one of the most widely used for video compression. [18] Not only does it allow the use of variable block size during motion compensation, which can massively increase the efficiency of modes P and B, but also specifies motion estimation with a quarter-pixel accuracy.In comparison to previous standards it can potentially reduce 50% of the bit rate. [1]

This codec subtracts the prediction from the current macroblock to form a block of residual samples. The predictive methods supported by the H.264/MPEG-4 AVC codec are more flexible

than in previous standards, allowing for more accurate predictions and, therefore, efficient compression of video. The residual sample block relies on Discrete Cosine Transform, using an entire transformation of 4x4 or 8x8 for the generating of a set of coefficients. The block of coefficients of this transform is quantised, meaning that each coefficient is divided by an integer value chosen for each macroblock representing a quantisation parameter. In the quantisation matrix the expected coefficients result is mostly or all zero. With more zero coefficients, the higher is compression with low coded image quality and with more coefficients different from zero, the better the decoded image quality with lower compression.

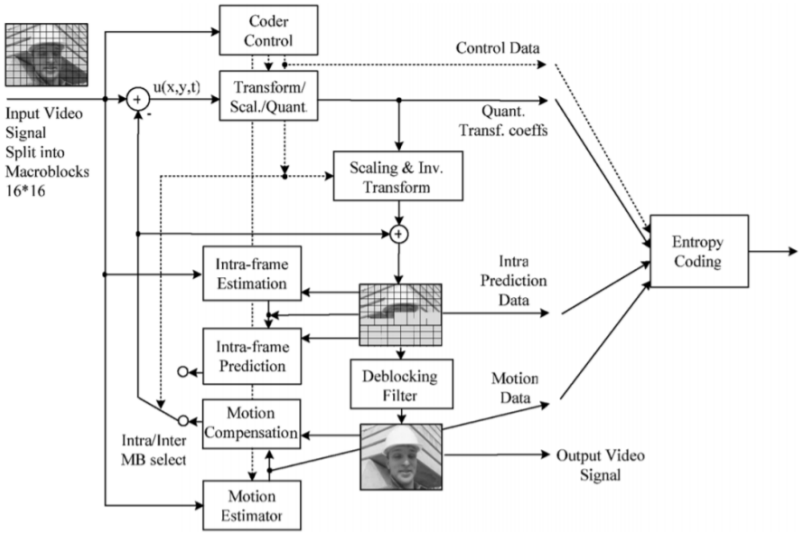

Figure 2.1: Block Diagram of an H.264 codec [1]

In the video coding process values are produced that must be coded to the bitstream decoding. These values are included:

- Quantified coefficients of the transformation;
- Information which allows the decoder to recreate the prediction;
- Information about the compressed data structure and compression mechanisms used during coding;
- Information about the complete video sequence.

.

During decoding processes, the decoder receives the bitstream encoding data and decodes each of the elements by extracting the information described above. This information will be used to reverse the encoding process and re-create a sequence of video images. The quantised coefficients of the transform are scaled and multiplied by an integer value to restore its original scale. The inverse transform will re-create each block of residual and combined samples will form

the residual macroblock. For each macroblock, the decoder forms a prediction identical to that of the encoder. Adds the prediction to the residue to reconstruct the coded macroblock to get a frame of the video.

2.1.2 High Efficiency Video Coding

High-efficiency video coding also known as H.265 and MPEG-H part 2 [19] is a coding method developed by MPEG and VCEG. This method aims to improve the performance of previous methods, such as the H.264 / MPEG-4 AVC. It is prepared to handle new resolutions, such as 4K, and parallel processing architectures. With better performance than the H.264 / AVC, the HEVC maintains the same quality but reduces bit rate by half [2]. An extension has already been developed for the HEVC, allowing it to support a multi-view system. [20] The encoder can achieve better efficiency using depth coding. Motion compensation is more efficient because instead of 8 directional modes used in the previous standard it uses 33. Replacing macroblocks for coding units will improve the performance of this codec.

The HEVC consists of coding a sequence of frames from the source video and creating a bitstream of the coded video. The bitstream coding is stored or transmitted and the decoder decodes the bitstream to obtain the sequence of decoded frames. HEVC has the same structure as previous methods. However many improvements have been implemented, such as:

- Prediction modes and transformation block sizes are more flexible;
- More flexible partitions, from large to small partitions;
- More sophisticated interpolation and Deblocking filters;
- Prediction, signage and more sophisticated motion vectors;
- Features to efficiently support parallel processing.

These points are based on [21].

Figure 2.2 represents a block diagram of the H.265 (HEVC) codec is presented. All resources involved in the process will now be presented and explained.

- Prediction Units and Prediction Blocks: The decision to encode an image area used for inter or intra prediction is done at the coding unit level. Supports multiple Prediction block sizes from 64×64 down to 4×4 samples;
- Transform Units and Transform Blocks: The prediction residual is encoded using block transformations. Whole-base functions similar to those of a discrete cosine transform are set for square sizes. For the 4x4 transformation of the intra-image residual prediction, an integer transformation derived from a form of is specified alternately;
- Motion vector signalling: Advanced motion vector prediction is used, including derivation of several most probable candidates based on data from adjacent prediction blocks and the reference picture;

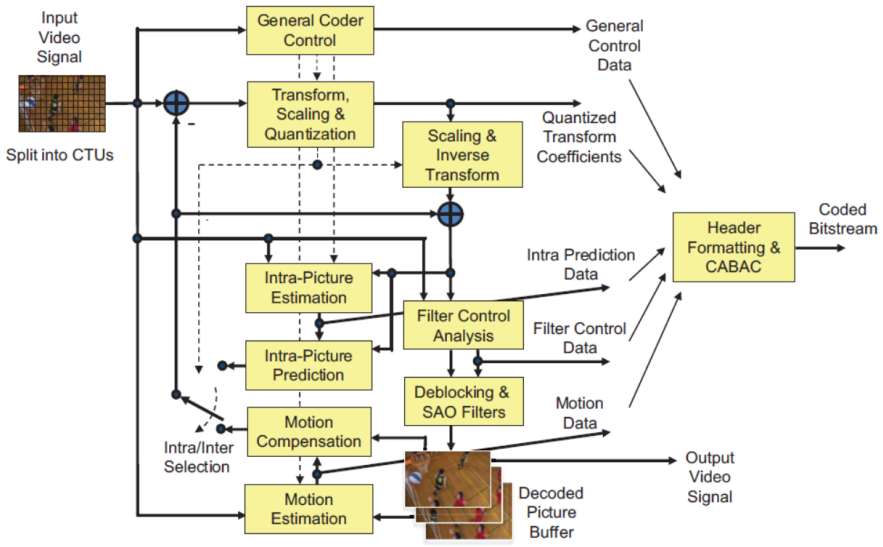

Figure 2.2: Block diagram of an HEVC codec [2]

- Motion compensation: Similar to H.264, several reference images are used. For each prediction block one or two motion vectors are used and can be transmitted either in uni-predictive or bi-predictive coding. The only difference is that it uses 7 or 8 taps filters for interpolation of fractional sample positions compared to 6-lead filtration in H.264;
- Quantisation control and In-loop deblocking filtering: These two processes are similar to the processes used in H.264;
- Entropy coding: Similar to the H.264 schema, but has undergone several improvements in order to improve throughput for parallel processing architectures, and their compression performance to reduce memory requirements;
- Sample adaptive offset: Adding a non-linear amplitude mapping in the inter-image prediction loop after the deblocking filter with the aim of better reconstruction the original amplitudes of the signal using a query table with additional parameters.

This section was based on [2].

The increase in processing power requirements in this new method also allows better data compression. The enhancements applied in this codec over the H.264 will provide:

- With the same size and image quality, the H.265 video stream should occupy less storage or transmission capacity than the H.264 video stream;
- With the same storage or transmission bandwidth the quality and resolution of the HEVC video sequence is greater than the corresponding H.264 video stream.

These points are based on [21].

2.1.3 DoF

The Degree of Freedom is an independent physical parameter used to describe motion in space, and thus obtain the various directions in which an object can move in three-dimensional space. In this dissertation will be studied the 3DoF, 6DoF and their comparison, in order to understand which is the best option.

2.1.3.1 3DoF

The Three Degree of Freedom explores three types of degree of freedom, which is translated by rotational motions in three-dimensional space, that is, rotation in the three perpendicular axes designating the type of rotation by pitch, yaw and roll. The pitch corresponds to the vertical axis, yaw corresponds to the transverse axis and finally, the roll corresponds to the longitudinal axis. In the virtual experience we can only detect the movement of the head, in this way we can look around but only from a fixed point of view of the user.

2.1.3.2 6DoF

The Six Degree of Freedom, is the free movement of a rigid body in three-dimensional space, so in six degree of freedom. These six basic forms of movement can be divided into two groups:

- Translation movements, that is to move up and down, back and forth, to the left and to the right;
- Rotation movements, that is the rotation in the three perpendicular axes designating the type of rotation by pitch, yaw and roll.

2.1.3.3 Comparison

- The 3DoF only allows rotation movements, while the 6DoF allows translation and rotation movements;
- The 3DoF does not allow having control of what goes on in the virtual experience. We can only observe from a fixed point of view, while the 6DoF allows interactivity in the experience;
- The 3DoF only detects the movement of the head, while the 6DoF detects the movement of the head and also the position of the user in space.

These comparison topics are based on [22].

After analysing the two, it is possible to conclude that the 6DoF offers a more interactive and complete user experience, allowing to not only visualise, but also to move around in virtual space.

2.1.4 MPEG I

Humans have sensors, such as eyes and ears. Through these sensors we can capture the physical environment. The information we get is a symbolic sample of the real world as we are limited by human sensors. With the advancement of technology and consequently with the evolution of sensors, the creation and usage of virtual environments is becoming very demanded, particularly in games or teaching.

These virtual scenarios are natural, captured from the real world, or synthetic, computer generated. In this type of experience immersion is fundamental and for this it is necessary to use all of the all available data in order to create a high quality environment very similar to reality. [3]

The key to these experiences is renderization. Rendering consists in a process that provides presentation creation. Its function is to imitate with high fidelity the reality extracting realistic patterns from the data. When rendering 6DoF content, it defines an initial scene and the client can choose between rendering the complete or simplified scene. [23]

This standard is based on the capture and creation of audiovisual information to create virtual environments or to improve immersive audiovisual experiences. [23]

2.1.4.1 Overview

In this section, the workflow of the sound and light field will be displayed from its creation to its display. Each phase will be studied in order to understand how the processing of this type of content works. Figure 2.3 shows the workflow.

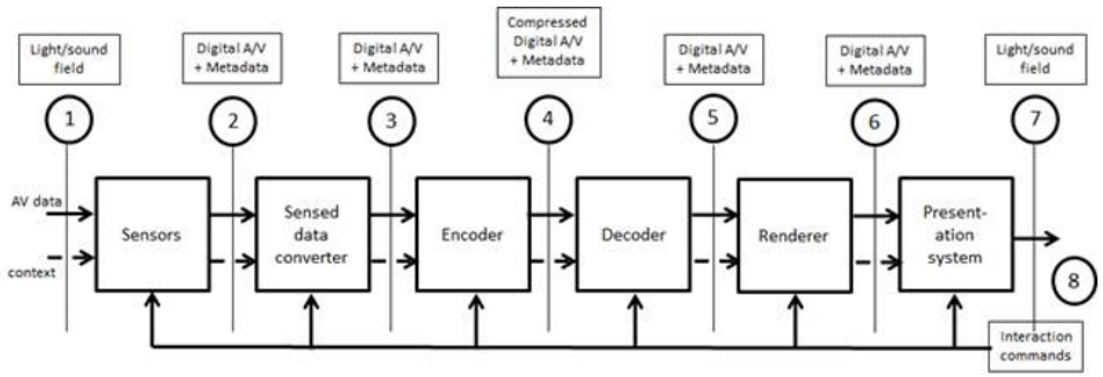

Figure 2.3: Light/sound field workflow [3]

These points are based on [3]

- Sensors - The first phase deals with the acquisition of data, through appropriate sensors. This acquisition of data can range from 2D cameras to depth-of-flight time sensors capturing natural content. However we can also create synthetic content through data generated by a computer. The sensor module may also include creation or capture of some metadata;

- Sensor data converter - The conversion module considers the application requirements in order to convert detected data into a more appropriate format, especially for interoperability reasons. Conversion can happen for both data and metadata;
- Encoder - The encoder aims to efficiently encode the data. This operation can be either with or without losses. The most common is with losses, taking into consideration that some of the losses are not detectable by human perception. This process applies to data and metadata;
- Decoder - The function of the decoder is to decode the data and metadata in the format of representation adopted after transmission or storage. Depending on the encoder, the decoded data will have a certain distortion compared to the original data;
- Renderer - The primary function of rendering is to extract the most appropriate decoded data for the best quality of the experience. The scene has to be extracted in order to obtain relevant data and this is always under user control. Rendering is a complex and sophisticated piece, it is what determines the quality of the experience, so it has a high importance. The metadata is processed as well;
- Presentation system - This module presents the final presentation of the content. Determines user experience and includes visual and audio components. In this phase the quality of the experience (QoE) will be evaluated, that is if the rendering played it is critical role, extracting the rich data;
- Command interactions - Scene representations have been improving to provide better interactivity to the user. Allows the user to interact with situations that happen in the real world. The commands that you can control are more diverse and can change the point of view or the focal plane. Over time, these interactions are expected to be closer to reality.

2.1.4.2 Use Cases

There are many use cases that span the light and sound fields in media applications. The following use cases are based on the technical report. [3] The focus will be on use cases intertwined with the theme of this dissertation, such as Free Browsing, where the user is free to select multiple views, and Interactive Reality, which implies the presence and interaction of a user in the content of the experiment, such as virtual reality.

- Omnidirectional 360°: The user is involved in a scenario that has been captured by a 360° sensor. The sensor is used to capture scenes and sounds of the environment. The user receives information from the perspective of the center of the content, that is, as if it were in the center of the environment, which will allow to observe and listen to the environment around. Through a mouse or head tracking the user has the possibility to change the view. After selection, the 360° content is extracted and rendered by the device in use;

- Free viewpoint live event: The sensors capture a closed or partially closed scene and therefore are placed around the scene. Several sensors are used, allowing multiple views of the scene. The user only sees the captured scene but can always rely on other live views or moments in the past. For example, a user is watching a handball game and wishes to review a previous detail;
- Free viewpoint cinema or television: The use case is identical to the live event of Free viewpoint, with the implementation that the monitor must be of high quality to be able to support all the details of the scene. The cameras capture multiple views and are placed around a closed scene in a studio;
- Remote surgery with glasses-free 3D display: The user puts on the glasses and is placed in a virtual reality environment. Uses a 3D model to perform depth, motion and space position detection. Detections can be used later. For example, a surgeon uses a 3D model to perform a virtual surgery, the movements performed are detected and subsequently executed by a surgical robot.

2.1.5 Alliance for Open Media Video 1

Alliance for Open Media is an organisation of companies that are contributing resources to create a truly open video codec. A codec that is not covered by high royalty rates seen in older encoders. The AV1 is a video encoder created by AOMedia, which comes to offer a decrease in the occupied bandwidth compared to popular encoders like H.264. The main purpose of this encoder is to offer high quality at lower bit rate. [24]

This new codec is covered by the following features:

- Open and interoperable;
- Optimised for the Internet;
- Designed to take up little space and optimised for hardware;
- Fits any modern device in any bandwidth;
- Ability to transmit video in real time, with consistency and in high quality;
- Flexibility in commercial and non-commercial content, including user-generated content.

These points are based on [25].

AV1 is a royalty-free solution that responds to increased demand for streaming high quality and fast. The AV1 encoder has five advances over the older encoders focusing on five main tools:

- Granular Film Synthesis;
- Constrained Directional Enhancement Filter;
- Distorted and global motion compensation;

- Increase in coding unit size;
- Arithmetic coding is not binary;

These points are based on [26].

Video is an essential part of our lives, from brief videos and video calls with family and friends, to the latest movies and news. Video is a media that allows us to share and unite and that can enrich our lives. Through collective experience, AOMedia has publicly made AV1 available, to help meet the growing demand for high-quality video streaming across modern devices around the world. In a near future, it is predicted its important impact on the world of video streaming. [27]

2.2 Networking

Computer networks are digital telecommunication networks that allow the exchange of network packets between computers all over the world, and this operation allows us to exchange information and resources. Nowadays, with the best quality of video production and the increasing demand for video streaming over the internet, it is necessary to apply transmission protocols that satisfy the needs of the user. In this section we will present some of the most currently used transmission protocols.

2.2.1 RTP

The RTP provides end-to-end delivery services with real-time features such as interactive audio and video, including in these services payload type identification, sequence numbering, date, time stamp and delivery monitoring. Applications typically run RTP over UDP in order to use their multiplexing and checksum services. However, the RTP can be used with other network protocols. Through multi cast distribution, the RTP can support data transfer to multiple destinations. This protocol does not provide any mechanism to ensure delivery on a timely basis or provide other guarantees of quality of service, but relies on lower tier services for this. The RTP allows the receiver to reconstruct the packet sequence of the sender through the included sequence numbers. These numbers can also be used to determine the proper location of a packet.

The RTP was designed primarily to meet the needs of multimedia conferencing, but it isn't limited in particular to that need. It is composed of two closely related parts:

- RTP: The transport protocol in real time, to transport data in real time;
- RTCP: The control protocol for supervising the quality of service and transmitting information about the participants in a session.

In audio and video conferencing, audio and video media are broadcast as separate RTP sessions. The RTP and RTCP packets are transmitted separately using two different UDP port pairs. A user who participates in both sessions so that they are associated, must use the same distinguished name RTCP packets for both. This separation allows some participants to receive only one media if they wish.

Multimedia applications must be able to adjust to the receiver's capacity or adapt to network congestion. Some implementations place this responsibility at source, but this does not work well with multi cast streaming. Responsibility for adaptation can be put into the receivers by combining a layered coding with a layered transmission system. In this context, the origin can distribute the progressive layers of a signal in several RTP sessions. The receivers can thus adapt and control their receiving bandwidth.

This section is based on [28].

2.2.2 TCP/IP

2.2.2.1 Overview

Generally, the term "TCP/IP" is related to the TCP and IP specific protocols, but may include other applications and protocols, such as UDP and ARP. The next figure is essential to understand the logical structure with which the TCP/IP operates. [4]

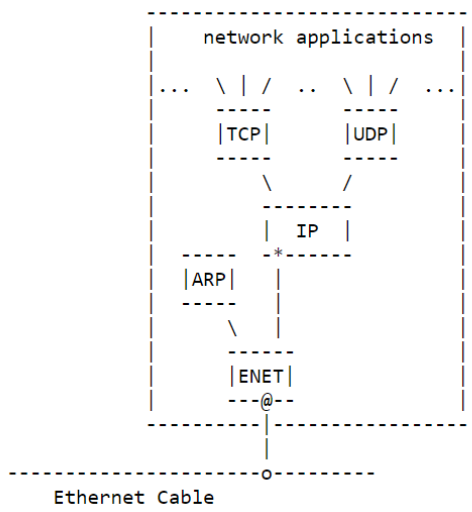

Figure 2.4: Basic TCP/IP Network. [4]

A computer can communicate using the internet, and in the figure is represented the logical structure of layered protocols of a computer on an internet. The boxes represents the data processing when they exit the computer and the connection lines shows the data path and the horizontal line represents the Ethernet. Both the TCP module, as the UDP or the Ethernet driver are n-to-1 multiplexers. It is possible to change many inputs to one output, but they are also 1-to-n multiplexers, and can change one input to many outputs, according to the type of protocol that is in the header. [4]

The packet can be passed to the ARP, which is used to convert IP addresses into Ethernet addresses, or the IP module if the Ethernet card enters the Ethernet driver outside the network.If an IP pact inserts IP, the value of the protocol field in the IP header will determine whether the data

unit will be transmitted to TCP or UDP. If the UDP datagram appears in the UDP, or if the TCP message appears on TCP, the message passes to the network application based on the port value in the UDP or TCP header, respectively. Data that comes through the TCP or UDP converges to the IP module. [4]

Multiplexing is simple to execute due to the initial point. There is only one path down and each protocol will add its information and header, so that the packet can be multiplexed at the destination. During execution a computer knows its own IP address and Ethernet address. [4]

2.2.2.2 Ethernet

An Ethernet frame contains the destination address, source address, type field, and data. An Ethernet address is 6 bytes. Each device has its own address and listens to Ethernet frames with that destination address. Ethernet uses Carrier Sense and Multiple Access with Collision Detection, so all devices communicate a single medium, which can only transmit at a time, but all devices can receive simultaneously. If two devices attempt to transmit at the same time, a collision is detected and both devices have to wait a period to try to transmit again. [4]

2.2.2.3 UDP

The UDP is one of the protocols that doesn't have space options to be a minimal addition to IP. Your header only provides ports and a data checksum for protection. [29]

This Protocol provides a packet-switched computer communication datagram mode in the environment of an interconnected set of computer networks, assuming that the Internet Protocol (IP) is used as the underlying protocol. The protocol is transaction-oriented, and duplicate and delivery protection is not guaranteed. [30]

The UDP is unique because it does not establish connections between the starting and the end of the communication system. The UDP communication can provide very efficient communication transport for some applications because it does not interfere with connection establishment and tear down overheads. A second unique feature of UDP is that it does not provide inherent mechanisms for congestion control, thereby describing the best current practice for congestion control on the Internet. A UDP datagram is carried in a single IP packet and is hence limited to a maximum payload for IPv4 and for IPv6. The transmission of large IP packets usually requires IP fragmentation, but fragmentation decreases communication reliability and efficiency, and for that reason should be avoided. [31]

UDP will add two values to what is provided by IP. One is the multiplexing of information between applications based on the port number and the other is a checksum to verify the integrity of the data. [4]

The information about the two values added is explained below:

- Ports: Communication between an application and the UDP is through UDP ports. These ports are numbered, starting with zero. The server waits for messages to enter a specific port dedicated to this service, waiting for any client to request the service. The data that is

sent by the UDP comes to an end as a single unit. UDP never joins two messages together or splits a single one into parts, the protocol preserves the message limit set;

- Checksum: An IP packet with a header type that indicates UDP, the UDP IP datagram examines the UDP checksum. If the checksum is zero, it can be ignored because it means it was not calculated by the sender. If the checksum is valid, the port number is examined, and if it is being used, a message application is queued for the application to read. Otherwise, the UDP datagram is dropped and will continue to drop UDP datagrams until there is space in the queue. If Ethernet is the only network between the two UDP modules communicating, then you may not need check summing.

2.2.2.4 TCP

The TCP provides a service other than the UDP because it provides a guided byte flow connection and guarantees delivery while the UDP does not guarantee it. Network applications that use TCP require guaranteed delivery and can not be disturbed with retransmissions or time-outs. File Transfer Protocol (FTP) and the TELNET are the two most practical applications that use TCP. These higher-capacity advantages of TCP have some costs, such as requiring more CPU and network bandwidth. [4]

Similar to UDP, TCP uses well-defined port numbers for specific applications. When the application starts to use TCP, the TCP module of the client and the server begin to communicate with each other. These two modules create a virtual circuit, where there is resource consumption at both points. The data can go in both directions simultaneously. [4]

The TCP is a sliding window protocol with timeout and retransmissions, with flow control at both ends to prevent buffer overhead. The window size is the number of bytes and not the number of segments, and it represents the amount of data that can be transmitted. [4]

2.2.2.5 IP

In the IP module, its essence is in its route table.This module has proved to be very important for Internet technology. IP uses this table in memory to make all the decisions about routing an IP packet. The contents of the route table are defined by the network administrator and any error blocks the communication. [4]

The IP module is fundamental to the success of the internet. As the message passes through the protocol stack, each module adds its header to the message and each module removes header as the message goes up through the protocol stack. IP header contains the IP address, which from multiple physical networks will build a single logical network. From this connection the name "Internet" is born. [4]

2.2.3 MPEG H

Multimedia services and the exchange of information have been explored by MPEG. The delivery of multimedia custom content for multiple users is done by the internet with more quality and in shorter duration. The MMT solved these problems when the preceded standards were not capable. I t consists of intelligent layers in the network next to the receiving entities, which store the content and pack the packets and send them to the receiving entities. [5]

The MMT is based on three requirements to reach the delivery service: the ease of accessing multimedia components, the reduction of the inefficiency brought by different formats and the easy combination for various contents. [5]

2.2.3.1 Overview

With the development of the internet, MMT brings an effective way to support the delivery of multimedia content. The MMT can be used in an IP environment and is composed of three main areas: the encapsulation, delivery and signalling, which will be described separately below.

- Encapsulation Encapsulation has the main function of processing the encoded content in a specified MMT format, thus defining the content structure; [5]
- Delivery: This area is responsible for defining the protocol that supports the delivery of multimedia content streaming. Also defines the format of the load to package the encapsulated data, taking into account the packet size of the delivery layer; [5]
- Signalling: Signalling is responsible for defining the message formats to control the consumption and delivery of the MMT package. The consumption messages signal the MMT structure and the delivery messages signal the structure of the payload format and the protocol configuration. [5]

We can see where these areas are inserted in the protocol through figure 2.5 where an example of the architecture of MMT is represented.

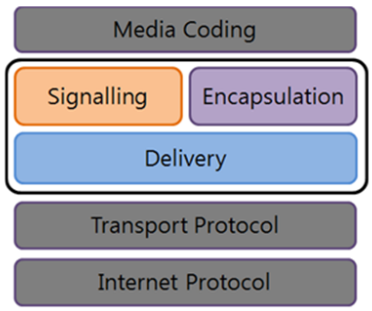

Figure 2.5: Architecture of the MPEG Media Transport (MMT). [5]

The content of MMT is composed of the Media Fragment Unit (MFU), the Media Processing Unit (MPU) and the MFU and MPU Package. To produce the MMT content, the data is decomposed into slices and the data that results from it is considered to be MFU. After several MFU's, these are combined in MPU and are considered for storage or consumption. Then one or more MPU's are combined into an Asset to provide multimedia components. The Individual Asset has its own identifier. Finally, the MMT package can be executed by one or more Assets and is represented by the spatial and temporal relationship between the Assets, and can be delivered by several networks. In short, an MMT packet is composed of elementary media streams and important data functioning as a broadcast program.This process was based on [32].

2.2.3.2 Use Cases

According to the type of experience involved in this dissertation and also the transport protocol used in it the demonstrated use cases are based on the immersive experiences and networks adaptability. The use cases presented are based on [33].

- 3D and 3D interactive video content: 3D content can pose a challenge to the efficiency of the transportation system. With the development of 3D technologies, and with advanced multi-view video encoding, transport is an increasing quality factor of the experience. For the user to enjoy these immersive experiences, he must obtain best quality possible for a better experience, such as 3D TV or 3D games. The same applies to interactive 3D and 3D video content;

- Adaptability in applications and transport: Each consumer has different characteristics, as such, to meet the need of each one of them are required multiple versions of the same content. Also, with the constant change of some parameters of the network, it is necessary to respond to these variations. There is a large amount of information about the content, server, network, terminal and user that can be used for adaptation. In the case of transport, the transport flow can help the adaptation, being able to discard packages according to the information provided, thus not allowing the filling of the buffer.

2.2.4 HTTP Live Streaming

HTTP Live Streaming provides an efficient protocol of long-term video transmission through the Internet, allowing the receiver to adapt the bit rate of the multimedia content to the network conditions, in order to maintain the reproduction without interruptions and in the best possible quality. Its structure is flexible for encrypting multimedia content, efficiently delivers multiple versions of the same content, is HTTP compliant, and still has a caching infrastructure to support large audiences. [34]

HTTP Live Streaming consists conceptually in three parts:

- Server: It is responsible for receiving input streams of multimedia content and digitally coding them;

- Distribution: This is a web server or Web caching system that delivers media content and content indexes to the client through HTTP;
- Client software: It is responsible for determining the appropriate multimedia content to be requested and for reassembling that content so that the user can view it in a stream.

A playlist of a multimedia presentation is specified by an identifier, called a Uniform Resource Identifier (URI). The playlist is a text file that contains the URI's, descriptive tags, and contains a list of media segments that, when played in sequence, will play the multimedia content. For playback of the playlist, the client must first download it, and after the playback plays each segment media. The URI can specify the type of protocol that can reliably transfer the intended resource, but they must be transported over HTTP. The more complex presentation is described by a master list that provides a set of variants, where each quay describes a different version of the same content. The variant stream includes media content encoded at a specific bit rate, in a specific format, with a specific resolution for media containing video and a set of renderings. The versions represent alternatives to the content, such as in video recording from different camera angles. The client must switch in different variant streams to adapt the network conditions. [35]

A playlist contains a series of media segments, which is specified by a URI and the duration of each media segment. Each segment has a unique integer sequence stream number. The number of the first segment is zero, and the sequence number of another segment is equal to the sequence number of the segment that precedes it, plus one. A media segment that contains video must include enough information to initialise a decoder and decode a continuous set of frames that includes the final frame in the segment. The more the information to decode the frames in the segment, the efficiency of the network will be optimised. [35]

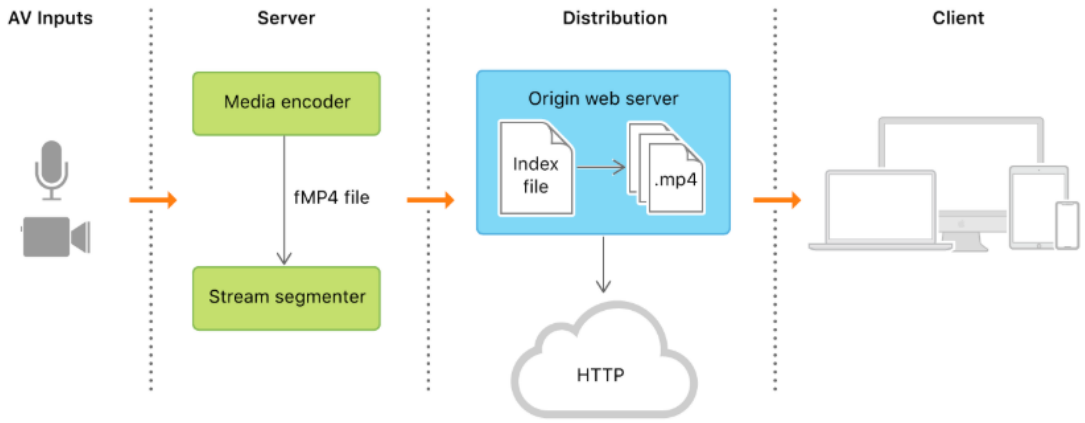

Figure 2.6: The components of an HTTP Live Stream. [6]

In a nutshell, the typical configuration consists of receiving an audio and video input into a hardware encoder that encodes video with the HEVC and audio and with the AC-3. Generates a fragmented MPEG-4 file or an MPEG-2 transport stream. The stream is divided into short

multimedia content, and placed on a web server. An index that contains a list of all media contents is created. The client software reads the index, requests the media contents listed in order, and displays them without pauses or intervals between segments. [6]

2.2.5 MPEG-DASH

With the evolution of electronic and computer technology, multi-view video transmission has been a focus of the research communities. Dynamic Adaptive Streaming over HTTP (DASH) has been gaining more clarity in the video streaming service. As such it has been studied for a better multi-view service to provide various points of view to the user. [36]

A multi-view video consists of a set of video clips with different viewing angles, which allows the user to select different viewpoints. This type of video can be used in many applications, such as broadcasting a sports game. Because there is a high number of web platforms and broadband connections, HTTP streaming has become a very effective medium for delivering multimedia content. [36]

DASH divides the video content into segments of equal length and stores them on a server. Segments are copied and coded with different bit rates, so they can offer different qualities and resolutions. Segments stored on the server can be queried in an XML-based Media Presentation Description (MPD) file. In MPEG-DASH, clients have control over the streaming sessions because the client can choose the requested bit rate, and a representation based on the condition of the network and its decoding process. [36]

DASH formats are designed for use with HTTP clients. First, the client retrieves a manifest in a Media Presentation Description (MPD) and only then based on that metadata does the client select, retrieve, and render content segments. [37]

When deployed over HTTP, DASH offers benefits over streaming technologies such as:

- DASH requests and responses pass through firewalls, just like any other HTTP message, because the content is stored on HTTP servers;
- DASH is highly scalable and can be stored in HTTP caches;
- A DASH client constantly measures the available bandwidth, and can manage the resources in a way to choose the segment that best suits its conditions based on this information.

This points are based on [37].

2.2.5.1 Overview

MPEG-DASH is based on dividing a file into segments that can be encoded at different bit rates or resolutions. Segments are provided by a Web server and can be downloaded over HTTP. In the following figure 2.7 we can observe the HTTP server serving three different qualifications, ie the same segment with the same duration, but with different types of quality. This adaptation to the

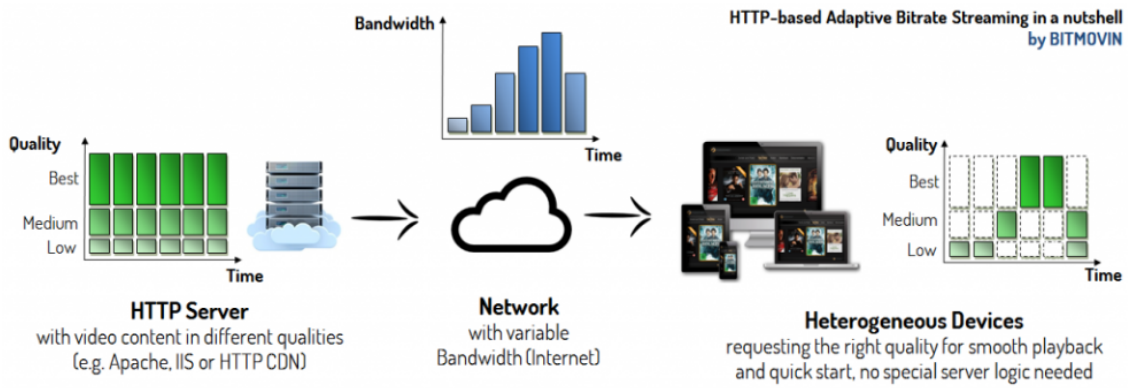

Figure 2.7: Representative Diagram of MPEG DASH [7]

bit rate brings several advantages to the customer, since it knows its capabilities and will allow the client to choose for each segment the bit rate that best suits its bandwidth. [7]

The Media Presentation Description was introduced in DASH, an XML file that represents the different qualities of the media content and the individual segments of each quality with its locator (URLs). This file provides other information, such as bit rate, start time, and segment length. The MPD is requested and presented first to the client in order to provide the necessary information for the customer to be able to request the segments that best suit their needs. [7] Alternative versions of media content are grouped in terms of codec, resolutions, or bit rate. Each alternative version represents one element. [38]

Finally, DASH-based multi-view video streaming suffers a delay that affects the quality of the experience (QoE), this delay is between a request for image exchange and rendering of the display. [7] This delay will be further studied in the problem characterisation section and when presented the proposed solution. [7]

2.2.5.2 Server

The server stores the three-dimensional content in the multi visualisation representation, where different views are saved. Each of these views will have an associated depth map video sequence, referring to the texture and depth of some view components. The sequences are then encoded at different bit rates to obtain different qualities using the H.264 or H.265 encoder for better compression efficiency. After coding, streams are segmented according to the DASH standard, rendering different representations with the same duration of the same content, but with different qualities, which means different bit rates. [39]

Storage requirements can be reduced, albeit at the cost of greater complexity of client-side decoding. In this system, the web server is an HTTP server that responds to requests from a DASH client. The HTTP protocol is used for communication between the server and the client, allowing caching of content transmitting, thus taking advantage of HTTP proxy caches. [39]

2.2.5.3 Client

The streaming client maintains four segment buffers and requests segments of four streams of video components according to the point of view desired by the user. The four streams you request correspond to two texture and depth streams for the nearest neighbour viewpoints. After decoding the received segments, the client renders one or two synthesised views. When the user requests a change of view, the system will gradually change the views, moving the viewpoint sequentially in the direction indicated by the user. [39]

2.2.5.4 Use Cases

The experts in MPEG derived some use cases, which are:

- Network mobility, for example when the user moves physically;
- Server fail over scenario, for example when the segment delivery node fails;
- Server-assisted DASH tuning logic, for example, when the server helps DASH clients select the most appropriate representations;
- Bidirectional suggestion between servers, client and the network;
- Support of real-time operational service through the user, in order to improve the overall quality of the service;
- Media synchronisation between devices.

These points were based on [37].

2.2.5.5 Spatial Relationship Description

Due to the increased demand for multi-view content and due to its high video size, mechanisms have been developed to provide efficient data delivery. Spatial Relationship Description is in charge of signalling the coordinates and size of the video block related to the coordinates of all video in the MPD. The client can calculate the location of each video block, and thus has the control of deciding to broadcast a small region which may consist of a few blocks or the entire video. Instead of being sent the complete video, the video is divided into a block matrix and only the blocks that are needed for the user's display are transmitted. [40]

Before the SRD, there was no MPD to associate spatial information. It was not possible to associate a descriptor with 360^{o} content, and, consequently, it was not possible to describe two videos that were spatially related to the same scene. The SRD tool is a new feature of MPEG DASH that expresses a spatial relationship of a video stream and another full-frame video. It allows for flexible scrolling and zooming without any conditioning on the video play. [41]

While viewing the video in 360^{o}, only a small part of the 360^{o} screen is displayed by the user. This part is equivalent to a specific region of the whole content, that is, they are spatially related.

In order to reduce the requirements of 360º video bandwidth, is used a technique that allows the recognition of the user's vision to transmit in the viewing window with the highest resolution of the content that covers the user's field of view, and transmit in lower resolution content outside the user's field of view. [42]

The main goal of MPEG-DASH is the interoperability between adaptive HTTP-based streaming solutions, defining the media presentation description model and the temporal one. In addition, the SRD supports spatial relationships between media objects that are generated from a source video. [43]

The MPEG-DASH standard has been enhanced with a feature that allows spatial access to streaming video. The SRD feature allows ad-hoc streaming of spatial sub-parts of a video to be displayed on a device that is supported by MPEG-DASH. They are considered spatial information that allows you to describe several videos that represent related portions of the same scene. This feature provides a variety of use cases, as described above, such as high-quality zoom in addition to conventional MPEG-DASH. [38]

2.2.6 Buffering Techniques

The multi-view video services have been extensively studied, in order to offer the best quality service to the user. In multi-view video streaming based on DASH there is a delay in switching views. The delay between the request and the view affects the experience negatively and consequently reduces the QoE. [36] DASH supports Video On Demand and live streaming. The multimedia content is layered and stored. Each block is segmented and encoded in different resolutions. The client requests the MPD from the server to obtain the information for each layer. [44] Some buffering techniques have already been developed with the purpose of minimising the delay in switching views. In this section we will study these techniques.

In the first case studied, the buffer is composed of three parts. One to control the occupancy of the buffer, the second to reduce the fill time, and the third to transmit the media segments continuously without the client request. The first studied system presents a parallel streaming, in which content segments are transmitted continuously without being requested by the client, in order to reduce the overhead in the transition of views. The client placing an order provides a set of information to the buffer so that it can choose the required segments that sends in parallel, in order to ensure the best continuity of video playback. In parallel segment streaming, several HTTP sessions are executed, which allows the client to request multiple segments. Whenever a view switch occurs, the buffer empties to respond quickly to this exchange. The buffer leaves only an offset to ensure continuous playback. After emptying the buffer, the client receives the segment of the view that has passed and the buffer reloads with the new data. These two steps allow the play of segments without interruptions, thus softening the latency problem in view switching. [36]

In the second case studied, the client maintains four segment buffers, each one of them represents a reference stream and transmits segments of a view. Each of the 4 buffers is responsible for a view and to ensure a smooth transition and to decrease latency in view switching the cameras that capture the scene are equally spaced. When the user requests a new view, since all segments are

synchronised for the same interval of reference, the process is instantaneous and simple, since it only needs to know which view is and the switching is performed quickly. If the customer requests a point of view outside the reference range, two reference flows are required. Coordination between buffers is required to transmit the requested view together and without delays. This method has proven to be effective in resolving the problem in latency while exchanging views. [39]

Due to the delay in view transition in a DASH based multi-view video system, it is difficult to offer a good QoE. With the development of this buffer mechanism, this disadvantage is extinguished, because they are efficient in solving the latency problem. The buffer engine allows seamless playback. This provides an immersive experience of better quality to the user. [36]

2.2.6.1 QoS and QoE

Currently, the internet is a dominant mean of communication and video has evidenced its prominence as an application. [32] Streaming performs with server-based streaming, but the advent of dynamic streaming brings a new method of content delivery and playback. This new dynamics has become a trend by bringing multiple benefits compared to classic streaming. These advantages will severely improve the quality of experience (QoE). It allows to adapt the control of the user, depending on both the device used for playback and the conditions of the network that it uses. A flexible service model and bit rates / quality can be dynamically coupled to changes in network and server conditions. [32]

Quality of service (QoS) in telecommunications networks is expressed in network parameters such as packet loss or delays. A good QoS does not imply a good QoE. QoS is objective and focused on delivering content efficiently while QoE is more subjective as it is subject to user feedback. However, DASH-based multi-view streaming suffers a long delay of view switching. This delay between views negatively affects QoE. A possible solution that has already been studied is the implementation of a buffer that during the streaming will prepare other views in advance, thus improving the flow of view exchange. The buffer will minimise the delay in the exchange and thus support the continuous playback of a multi-view video. [36]

QoE is the parameter that defines streaming performance, which must be maximised in order to provide the best user experience. Ad-hoc streaming has improved the QoE quality of video streaming.

2.3 Eye Tracking

2.3.1 Introduction

Eye Tracking lets you observe and record the movement of a person's eyes. With an Eye Tracking technology, it is possible to obtain information through sensors applied in the user's face. This class allows you to detect a person's presence, focus, and other mental states. [45] This technology has a great potential, due to its wide variety of applications, being able to perform tasks only with the look. It has been a plus in the business world and health for providing information on

people's interpretation. It brings more advantages such as better interaction with computers or other devices, and can be used as a means of communication.

This chapter contains content on the methodology of Eye Tracking, some applications, and the eye tracking applied to machine learning.

2.3.2 Methodology

Eye tracking is done through ocular screening factors such as corneal reflection and the center of the pupil. All stimuli are recorded by the eye tracker to describe the movements made by the eyes. The eye tracker emits infrared rays into the user's eyes, which will strike the pupil and return to the device. Infrared light is used for better user comfort. This process allows you to identify where the user is looking. After the application detects the location of the corneal reflex, we can obtain the focal point of the user so that he can know where the visual attention of the user is. [46]

The movements of the eye is detected through the following systems: Mechanical, electronic and video systems.

In mechanical systems, for example in the use of contact lenses, the process involves the use of a contact lens with mirrors covering the eyeball. These mirrors have the function of detecting movement. Although the high precision of this method is not generalised because of being an invasive method and because the head could not be moved. [47]

In the electronic systems, through the use of an Electrooculogram we can measure the biopotentials of the spheres of the eyes through electrodes. Electrodes are placed close to the eye because the movement of the eye causes the surrounding electric fields to move and the voltages of that movement can be measured by approaching the detector of the eye. Although it is an efficient method, it has a cost of signal amplifiers,and the annoying presence of sensors that are applied to the user's face. [47]

In video systems two types of images are used, images in the visible spectrum, and images in the infrared spectrum. In the visible spectrum, the results depend on ambient light, which means that the lower the light, the worse the efficient detection. In the case of the infrared spectrum, the eye is constantly illuminated, thus eliminating the problem of ambient light. The camera should be placed below the visual axis of the eye and requires the presence of a user and calibration at the beginning of each experiment. The disadvantage of this type of method is the non-control of the head position. One way to eliminate this disadvantage was to use an eye tracker because it is a system that offers more mobility, good performance and does not offer high costs. [47]

There are two techniques that can be used to adapt to different environments and solve the problem of poor lighting. These techniques are bright pupil and dark pupil. Bright pupil is a technique used in artificial light conditions, such as indoor environments and the dark pupil is one used in outdoor environments. [46]

There are two types of eye-trackers, the intrusive and the non-intrusive. The intrusive ones are those that the participant needs to carry a device of his own, but offer more freedom of movement, such as glasses. Non-intrusive ones are those that record distance data and offer a more comfortable experience. Eye trackers need to be pre-calibrated to fit the characteristics of each person.

This process is called Calibration. The process is based on a point displayed on the screen and the user has to focus at that point for a preset time. This procedure aims to identify the center of the pupil and the cornea-reflex relation and record its position in the three-dimensional space. [46]

2.3.3 Metrics and Applications

Eye tracking methods have been sought after as technology progresses. Many studies seek to use this technique in a convenient and natural way. Ocular movement is an interaction tool in these applications. Eye Tracking has already been used in areas of psychology and neuroscience, for example in autism or Alzheimer's disease, schizophrenia or dyslexia, in industry or education. [47]

Eye tracking systems have metrics relevant to applications. Each metric satisfies a task, as are several metrics that satisfy multiple tasks. The following metrics are highlighted:

- Fixation: Refers to the moment when the eyes are fixed and to assimilate information or process the image; [47]
- Saccade: Refers to the moment in which the eyes change from a fixation to another fixation, that is to the interval of time between two fixings; [47]
- Gaze duration: Successive fixations in an area of interest. It is measured by the duration of the look, which results from several fixations. When the fixation occurs outside the area of interest, it marks the end of the look; [47]
- Scanpaths: Its main objective is to indicate the transition between areas of interest and thus provide information on the efficiency of the layout of the elements; [46]
- Area of interest: Visual area that is of interest to the entity doing the research or study and not defined by the participant. [46]

As this technique has several metrics through the study of human eye movements, and thus can understand how they interpret and process the environment around them. Eye tracking can be applied in education, psychology and neuroscience.

2.3.3.1 Education

Eye Tracking is used in education to study learning processes. It has proved to be a very effective tool for designing, evaluating and improving education. Visual attention has been a focus of research by many researchers in order to be analysed and applied in education.

Eye Tracking can help us discover information we get or how educational material works. It can also detect learning strategies, social interaction and skills used in problem solving. At the educational level it is a tool used to understand the learning process, the load and the learning methods, being able to obtain information for improvements in these processes.

This application was based on [48].

2.3.3.2 Psychology and Neuroscience

Eye tracking is used in the fields of psychology and neuroscience in order to understand the connection between what we see and how we react to the information we process.

Eye tracking in the cognitive process is used as a tool that allows us to study human behaviours such as attention, memory or perception. This technique allows to understand the mental processes realised by a human during an activity.

In social psychology this technique is used as a tool that measures the influence of human behaviour through the visual impact provoked, allowing to evaluate data such as social influence, persuasion or group dynamics.

In neuroscience the eye trackers are used to understand neurological function and processes. Many researchers use eye trackers to investigate the development of brain damage, neurological diseases, or visual functions.

This application was based on [49].

2.3.4 Machine learning using eye tracking

Machine learning studies how to automatically learn to make an accurate forecast based on previously available data or observed data. The main purpose of Machine Learning is to provide highly accurate prediction of test data. Applying fully automatic methods, with the crucial incorporation of knowledge of human beings. This process brings advantages, such as being in some cases more precise, does not require a special type of programming, is an automatic process and has the peculiarity of being cheap and flexible, and can be applied to any learning task. However, it does have some drawbacks, such as needing lots of labeled data and is influenced by errors as it is impossible to achieve perfect accuracy. [50]

For a better understanding we can group machine learning into three groups. These groups are:

- Supervised Learning: In the supervised learning algorithm the model goes through a training process where the predictions are made and this process is only completed when the. The template is corrected when the results do not match what you expected. The input data is called training data and its result is known.This type of algorithm are used in classification and regression problems;

- Unsupervised learning: In the non-supervised learning algorithm the model is prepared by deducing structures present in the input data to extract general rules. The input data is not labeled and has no known results. This type of algorithm are used in clustering, dimensionality reduction and association rule learning problems;

- Semi-Supervised Learning: In the semi-supervised learning algorithm the model has to learn the structures to organise the data and make predictions, although there is a desired forecast. Input data is a mixture of labeled and unlabelled. This type of algorithm are used in classification and regression problems.

These points are based on [51].

In recent years, Eye Tracking has offered an alternative to conventional communication modes. The eyes reflect the mind, they provide information about thoughts, intentions, mental states and our focus of attention. [52] Through the eyes we identify what interests us. The look estimation can be used to interact with a computer to help people with motor disabilities. It can also be used for robotics. In health it also has influence and can help to diagnose diseases or psychological evaluation. [53]

A new structure uses the look as input patterns in order to carry out activities, predicting the user's intention and thus helping to overcome difficulties or to perform tasks. This method consists of two modules. The first module is composed of regions of interest and the second module corresponds to a convolution neural network (CNN) that is trained and used for recognition. The prediction is performed with assistance in the characteristics of visual attention by collecting information on the number of fixings, duration of fixations, and the path that the eye follows. [54] The efficiency of the detection algorithm and the robustness of the process, are two essential points for a good implementation of this technique. [53]

This technique that helps to interpret the user's intention and focus of attention is used in multiple applications, such as robotics, in health. For example, the use of eye movement data is associated with the development of assistive technologies for paralysed patients. Patients have extremely limited motor and communication skills, but it does not affect their cognitive abilities and their ability to move their eyes. As such, Eye Tracking can be used to create assistive technologies for such problems, such as through other forms of communication. [55] Technologies are also being developed to help limited mobility cases through robotics. An example of such application is in the visual orientation of a child with mobility limitations it can provide information to the robot about which toy the child wants to play with. [52]

All this technology allows us to carry out simple activities, offers security, and improves the quality of life of some people with disabilities. It has become a plus in our daily lives through success in problem solving. [55]

Chapter 3

Methodology

This chapter presents the methodology adopted for this dissertation, including its objectives and the chosen approach to solve the presented problem. In this chapter, we also describe the work plan, documentation, work platforms and tools used to develop this experiment. Finally, a description of test scenarios and experiments' conditions are presented.

3.1 Objectives

In order to complete the main objective of this dissertation, the following intermediate steps had to be taken into account:

- Clear identification of the key challenges presented when transmitting multi-view or 360-degree video over shared IP networks;
- Understand the use of technology to represent, transmit and watch multi-view and / or 360 video, especially the MPEG-DASH specification and associated reproduction and transmission technologies;
- Clear identification of key aspects that primarily contribute to deteriorate the quality of multiple views or 360-degree streaming applications' experience; especially, those concerning the latency whenever the user changes his attention focus, which according to previous works is essential;
- Define an approach to establish a correlation between the viewer's attention focus and the representation of multi-view content on the server side according to the MPEG-DASH specification;
- Identify appropriate approaches to minimise key identified challenges and specify the corresponding mechanisms;
- Development of proof of concept and test the developed mechanism.

3.2 Approach

The current version of the prototype has some limitations, one is offering reduced quality when switching views and other concern is adapting instantaneously to the user's focus of attention. An overview of this Prototype is described in chapter 4.

Initially, tests were performed on the Prototype to obtain the current status of latency and performance parameters during view switching. The data obtained in the tests provided relevant information concerning the level of quality in terms of QoS and QoE that the existing Prototype is able to deliver. The tests thus enable to establish the baseline against which the results obtained in this dissertation will have to come forward.

The tests performed in the Prototype reveal a latency problem when a new view is requested. In order to achieve a functional system this problem must be overcome and for that was designed a Proxy with a buffer mechanism responsible for the view switching operation. The aim is to improve the parameters of quality and performance of the system with this new mechanism. The mechanism developed is described in chapter 4 on section 4.6.2.

To test the Proxy a new test Prototype was created and tests were performed to obtain latency and performance parameters during switching of views. The new Prototype for test is described in chapter 4 on section 4.7. These tests were essential to compare the current state with the new one using the new buffering mechanism and to evaluate the viability of the work done.

As the Proxy was not incorporated in the Prototype a suggestion for incorporation was described in chapter 6.

3.2.1 QoS Parameters

The initial Prototype should ensure that QoS is offered and to do so the priority was to solve the latency problem during view switching that affected QoS and, consequently, also QoE. In this Prototype are identified two types of latency that involve the switching of views:

- Switching Latency: Latency that exists between the request of the view and the reception, that is latency since the request of a new view from the head tracking until the moment it is presented to the user;

- Head tracking latency: The latency that exists in head tracking, from the moment movement is detected until the view is requested, that is, latency between motion detection by the webcam and the request for a new view.

3.2.2 Work Plan

During this dissertation, a work plan was followed. The figure 3.1 illustrates this plan.

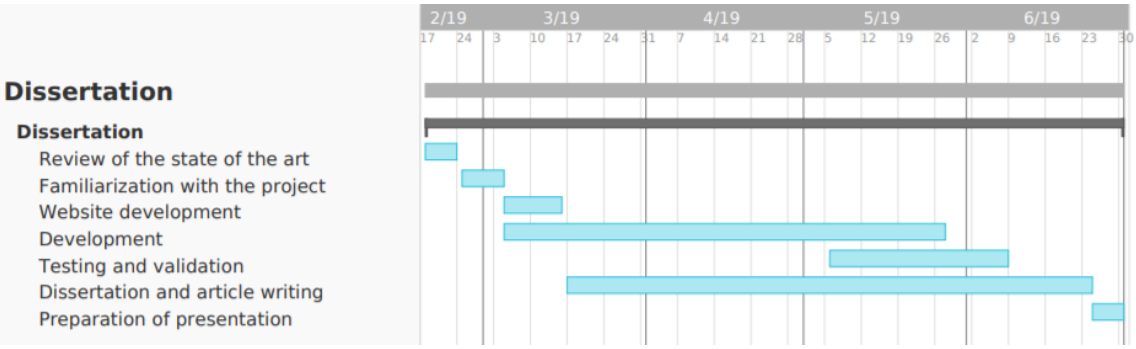

Figure 3.1: Work Plan

- Revision of the State of the Art: The State of the Art was initially reviewed in order to guarantee a total knowledge about all the concepts and technologies that would be approached. During the writing of this dissertation the State of the Art was updated in order to guarantee that all information was collected;
- Website development: Time period reserved for website development;
- Familiarisation with the project: Due to the complexity of this project, a period of 2 weeks was reserved for adjustment to the Prototype and its tools. During this period, the Prototype functionalities were studied in order to better understand the project, and also to find the better approach to achieve the presented objectives;
- Development: In this part, a mechanism was developed to achieve the objectives of this dissertation within the foreseen period. The development and implementation of this new mechanism was achieved during this period;
- Testing and validation: This step was reserved for performance testing in order to obtain an evaluation of the developed mechanism. The tests were running while part of the writing was done;
- Dissertation and Article writing: Period of time reserved for writing this document and for writing the article;
- Preparing Presentation: Period of time reserved for preparing the dissertation's presentation.

During the development of this work, there were some delays that were overcame with parallel work, so that the project was finished within the foreseen period.

3.2.3 Technologies,Tools and Work Platforms

Considering that the mechanism in development in this dissertation would have to inter-operate with an existing prototype, special attention was given to the selection of tools and technologies to ensure compatibility and smooth integration.

The information provided by the client is detected manually by means of a GUI client written in Python. Python was chosen because of the ease in programming language and because the GUI is only going to be used for testing and will not be incorporated into the initial Prototype. The GUI is available for Windows platforms.

> ” Python is a programming language that lets you work quickly and integrate systems more effectively” [56]

The Proxy that manages the requests from clients and the responses from the server is programmed in C language such as the Server. These two models are available and were developed in a Linux environment.

3.2.3.1 Ubuntu OS

The initial Prototype is created in C / C ++, and to achieve the objective of this dissertation it was necessary to develop the algorithm in the same way. The Ubuntu operating system is able to provide toolchains that allow to develop and compile C code efficiently and quickly. In order to do so, the GNU C compiler was chosen as it allows for an easy and flexible development under UNIX based operating systems.

> ” Ubuntu is an open source software operating system that runs from the desktop, to the cloud, to all your internet connected things ” [57]

3.2.3.2 Oracle VM VirtualBox

A network system between a Client, a Proxy, and a Server was designed and two virtual machines were created using the Ubuntu OS, one of them representing the Proxy and the other the Server. The machines are in Bridged Adapter mode and communicate via sockets following the HTTP protocol.

> ” VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on ” [58]

3.2.3.3 Documentation

This dissertation and respective article were written and developed in LaTeX using Overleaf, which is an online LaTeX editor. [59]

3.3 Test Scenarios

The tests performed in the developed experimental Prototype intended to evaluate the efficiency of the Proxy. The tests also intended to evaluate the performance of the Proxy, so as to show its good operation in the production environment.

The goal of the tests is to measure latency in the view transition. The values obtained will be compared to the latency values obtained without the new buffering mechanism for view transition. For this test, it must be taken into account the initial state of the Prototype and for that the latency values collected in MSc thesis [60] related with the previous work are shown in the table 3.1. This values are related to the two types of latency in the switching view that were mentioned on subsection 3.2.1.

Table 3.1: Latency of view switching in the initial prototype

Latency - View switching			
Switching Operation	Left-Right	Left-Center	Center-Right
AVG. Latency(s)	2.5	2.5	2.5

Three machines were used to test the performance of Proxy in the new test Prototype: one for the Client, another for the Proxy, and finally one for the Server. The new test Prototype could be tested using two machines but in order to obtain better processing performance it was tested with three. The characteristics of the machines used for testing are presented in the table 3.2 .

Table 3.2: Machines specifications

Function	Client	Proxy	Server
Device	Toshiba	Lenovo	Lenovo
OS	Windows 10 Pro x64	Elementary 5.0	Lubuntu 19.04
Processor	Intel i7-4799MQ 2.40 GHz	Intel i7-8550U 1.8 GHz	Intel i7-8550U 1.8 GHz
Memory	8 GiB	8 GiB	8 GiB

The Network topology used for tests is based on a star topology, which means all the nodes are connected by a central node. The central node controls all transmissions, because the nodes connect through it. The figure 3.2 illustrate the Network topology that was adapted for the tests and experiments.

The tests had the following steps:

- In the first step, the data about total transition time of the view would be collected. This time interval represents the time required to achieve a high quality view requested by the Client;
- In the second step, view transition time data would be collected. The Client requests a new view and the Proxy sends a low-quality view while requesting a high-quality view;
- In the third step, it would be collected the time period since the request and the reception between the Client and the Proxy;
- In a fourth step, it would be collected the time period since the request and the recapture between the Proxy and the Server.

With the collected data it was intended to reach the following objectives:

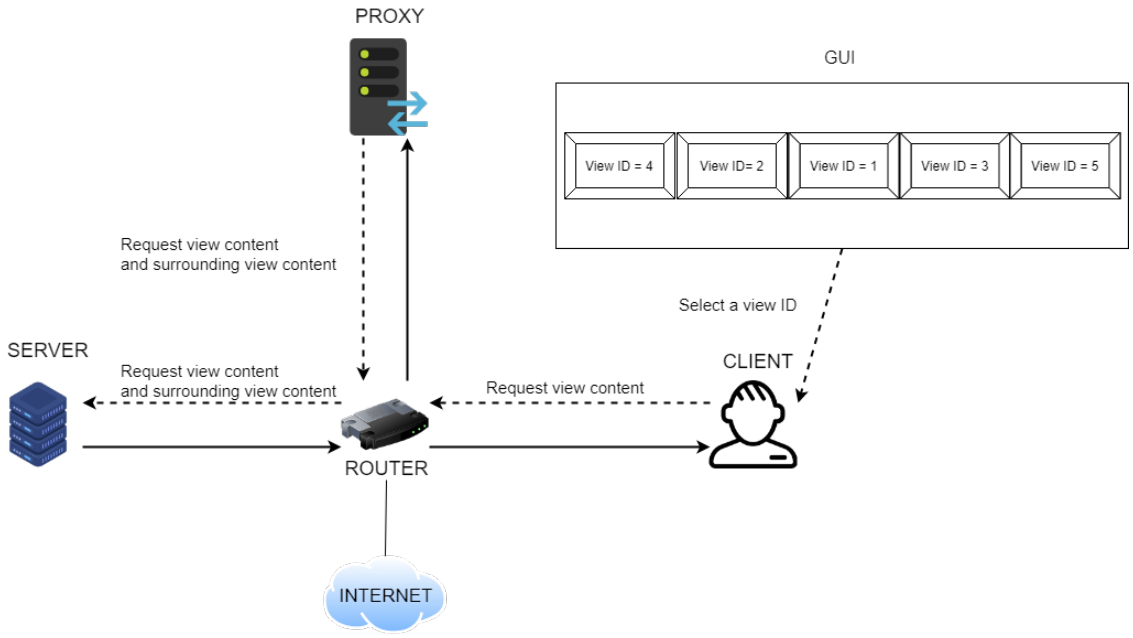

Figure 3.2: Network topology

- Determine view transition time: the time it actually takes to switch the view to a low quality and the time it takes to obtain a high quality of the same view;
- Realise which transition takes longer: collect the sub delay time between the three modules, in order to realise where the largest delay occurs.

During the tests it was taken into account an initial situation that was always respected in all tests. The change of view can only be requested after the initialisation files have been sent and received by the Client.

These tests aim to prove an increase of the multi-view experience's QoS and consequently of the QoE. These were the conditions that had been provided and were necessary for testing Proxy performance.

Chapter 4

Overview of the developed Solution

This chapter is designed to describe the Prototype used as starting point for this dissertation. It will be described its architecture and all its components. A buffer layer that contains a Proxy with a buffering mechanism will be presented. It will also be described a new test Prototype developed for testing Proxy performance. Its architecture, switching view operation and components will also be explained in detail. A Use Case is described in order to understand how to relate this work with a real situation. This chapter also has a list of assumptions to follow for the smooth operation of the experiment.

4.1 Architecture

This diagram in figure 4.1 illustrates the architecture of the prototype that was used as starting point for the development of this dissertation. Below, some aspects of its function layers will be explained.

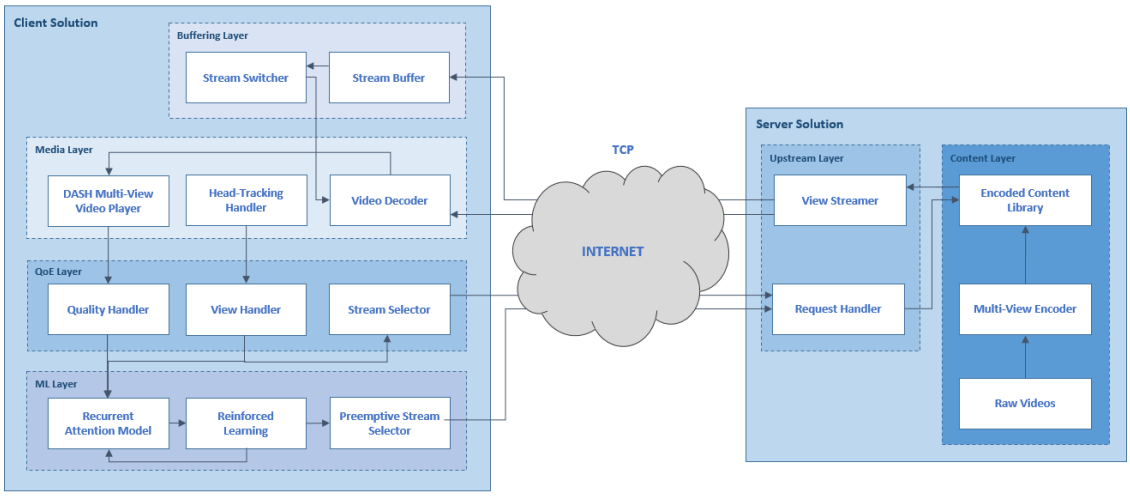

Figure 4.1: Architecture Diagram [8]

The Client subsystem, which consist of a Player and a Client Application, controls the streaming flow and the prototype behaviour. The Player is responsible for switching operations requested by Client, and does that communicating between systems. The Client Application has a subsystem with head tracking that supports the view switching operation, but it can also do it manually. The Player has access to the metadata with the performance and network information. The view quality is adapted by Client according to the available bandwidth by monitoring QoS parameters. This adaptation can also be done manually.

HTTP Server is composed by MPD and by the storage Server with all media streams on a single server distribution. The MPD hosted in Server contains a description and URLs for media streams hosted in storage Server. With a good communication between systems and subsystems the prototype provides a good QoS, seamless streaming and good QoE in different environments and conditions.

4.2 Server

The Server previously used in this Prototype only used one socket to communicate, following the HTTP protocol. It contained the MPD file and the media segments. This Prototype deals with problems when switching view operation is performed and to overcome that problem it's necessary to communicate through additional sockets. Those sockets enable the code to parallelly received the surrounding views that provide a smoother experience during the switching view process. This way it will be provided surrounding views which will help solving the switching view latency problem. The figure 4.2 represents MPEG-DASH standard structure illustrating its functionalities.

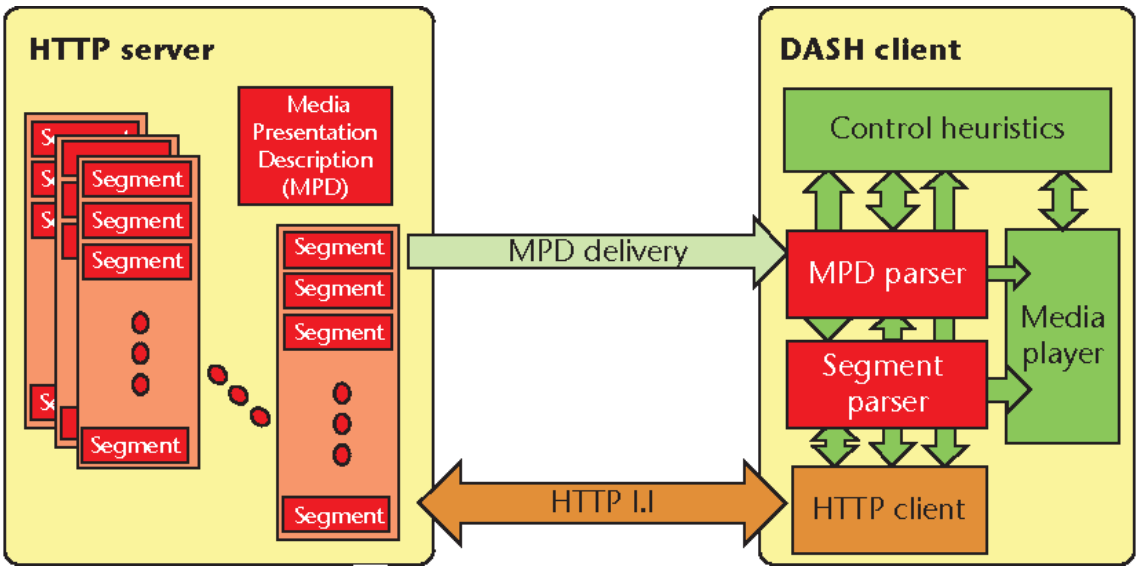

Figure 4.2: MPEG-DASH standard structure[9]

4.3 Media Streams and Media Encoder

This Prototype, based on VOD applications, generates media streams using MP4Box provided by GPAC [61] framework. This tool prepares multimedia files that support the MPEG-DASH specification. After the content is generated by MP4Box [62] is stored in HTTP Server.

The media content consists of a five view shoot of a concert in Universidade Católica with an array of 2D cameras placed equally distant in arch and also equally distant to the object of interest. This way it was obtained five different views of the same scene: Most Left, Left, Central, Right and Most Right, each one of them coded in four different bit rates obtaining this way four different quality modes. All the segments of the representations were coded using MVC, a multi-view extension of H.264/AVC standard. The initial segment in MP4 format and the others in M4S format.

4.4 Client

Client consists of two modules: The Client Application and MPEG-DASH Player. Client is responsible for parsing the MPD and for selecting the representation that better fits its needs.

Client controls the content adaptation to the bandwidth available and according to its fluctuations it is responsible to choose the appropriate segment representation. The multimedia player used is MP4Client provided by GPAC [61] and it can support the MPEG-DASH and others streaming techniques [63]. MP4Client includes a GUI and is also responsible for streaming content.

MP4Client is configured according to a configuration file. It supports five different views and this way can support the switching view operation commanded by Client. Client application and MPEG-DASH Player connect providing a interface for switching view manually or by head tracking, and also manually switching quality operation.

Client is capable of providing five different views and four different quality modes and also supports three view sets: linear, cross and cardinal.

4.5 MPD

MPD is an XML file which is essential for proper switching. The storage and mapping of each content is necessary to provide content quickly and efficiently. Each view is organised and represented in MPD.

The MPEG-DASH profile used is On Demand and the files were delivered by HTTP. All the content storage in the Sever is represented in MPDs, which provide all available content.

After the Client parses the MPD, stream technical characteristics and all available representations of the content will be provided to it. The provided content in the storage Server was collected from a five different views shoot of a concert in Universidade Católica.

Each representation has an initial segment and 1009 media streams for four different quality video views: very low, low, medium and high quality. The media streams has one unique URL

that provides the exact location in the server storage. The MPD has various representations of the same content. Every different view has the same number of segments in order to provide a continuous flow while switching view and this way maintaining time synchronisation. Each view has 1010 segments aligned in time, the first one is the initial segment and has the mp4 file type and the rest of the segments have the m4s files type with the duration of 119 ms. The figure 4.3 illustrates the MPD file structure.

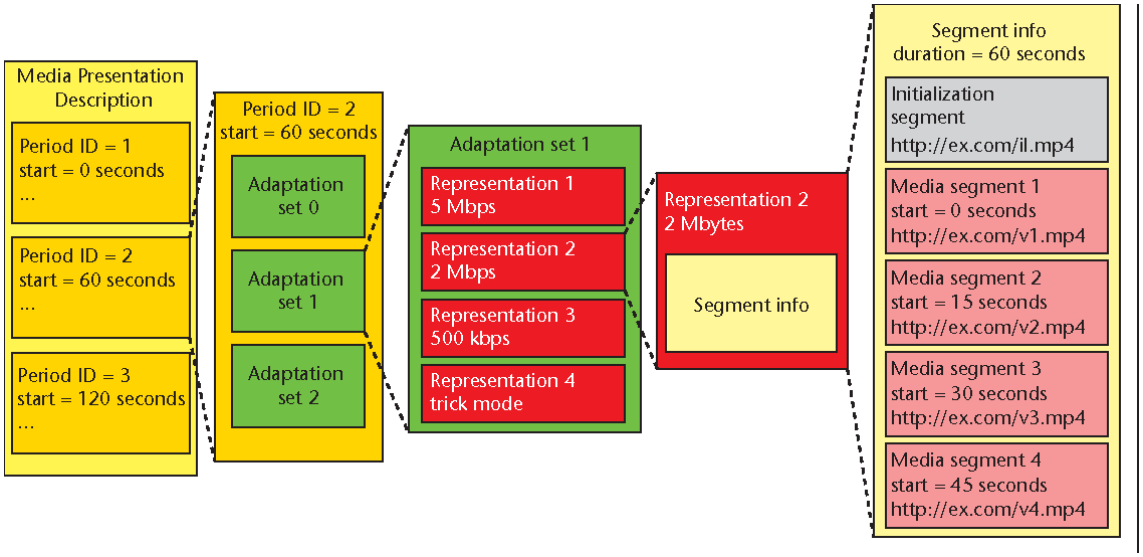

Figure 4.3: Scheme of MPD document structure [9]

The initial MPD described only segments for Central, Left, and Right view. It only can parse these three views, which is a limitation for this experience. In order to achieve more views and different ways to test the view transition, two more views were added to the XML file: Most Left and a Most Right view. This change was made in order to improve this experiment with a more number of views and to show that Proxy is able to provide additional flexibility to the existing Prototype.

4.6 Buffer Layer

In this section, it will be explained one of the main objectives of this dissertation: development of a buffer mechanism. The planing for the creation of this mechanism and its incorporation in the Prototype was done through a Proxy. Below, it will be described the Proxy and the Buffer mechanism developed.

4.6.1 Proxy

Proxy communicates with the Client through a socket and communicates with the Server through four sockets, following the HTTP protocol.

When the Prototype is initialised, the Proxy requests the MPD to the Server, and the experiment only begins when that request is completed. After received, the MPD file is parsed once and it's stored in memory for additional use. To execute parsing operations, the expat library was used. [64].

Then, considering that the central view is the initial one, the Proxy will analyse the view ID correspondent and will send the URLs associated to the view request and surrounding views. This process is repeated for every switching operation that is explained in the section 4.7.4.

Proxy keeps the received files in queues. It has four queues: one for central view, which corresponds to high quality views, two queues for lateral views which are low quality views and an auxiliary queue used in the buffering mechanism that is explained in section 4.6.2. Proxy will fill lateral queues with the surrounding views and the central queue with the Client view request sent by the Server. The files in central queue are the only ones that are sent to the Client.

Finally, Proxy will provide the time period between the request to the Server and the response. This type of data is important to have insight about sub delay in the free modules which provides information about its performance.

4.6.2 Buffer mechanism

This work addresses a buffer mechanism, which will be explained in detail throughout this section. The Buffer is composed by a system of queues based in FIFO algorithm. The figure 4.4 represents how FIFO algorithm functions.

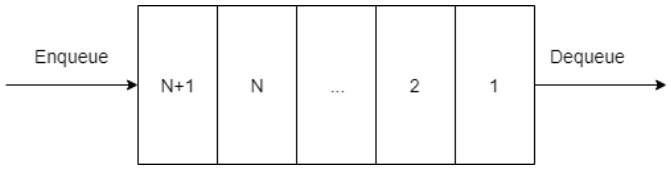

Figure 4.4: FIFO Architecture

In Proxy, as mentioned and described before, were created four queues. When the experience begins, the two lateral queues are filled with the files of the surrounding views sent by the Server. When the experience begins the Proxy starts to send the files from the central queue. After sent a file, it will do dequeue of the file on central queue and both respective files on lateral queues and enqueue a new segment in each respective queue. During the process, without view transition the auxiliary queue is empty. The figure 4.5 illustrates the process.

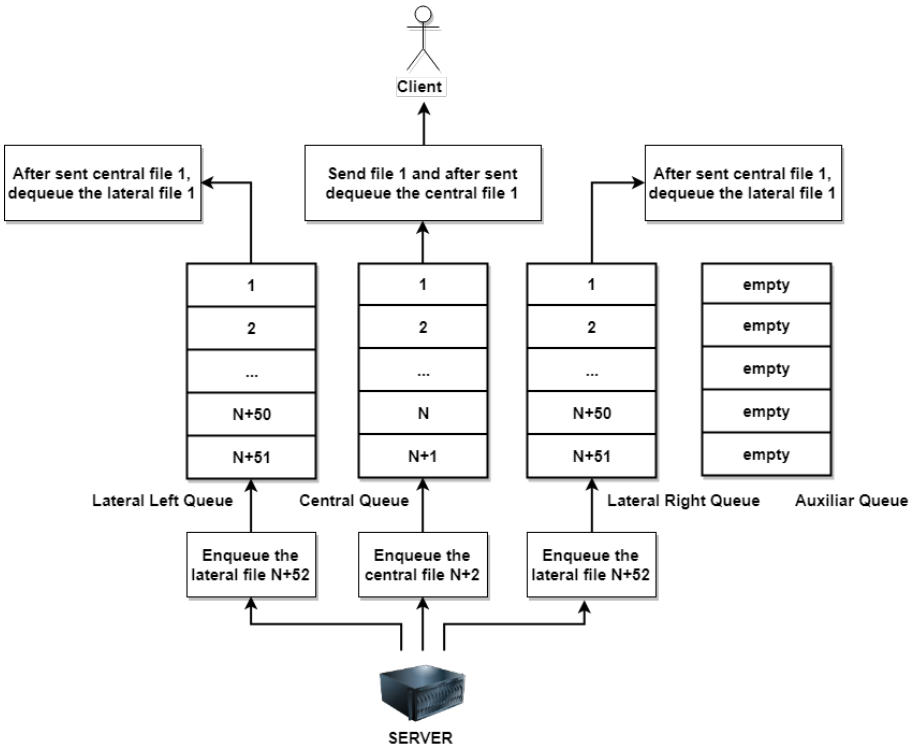

Figure 4.5: Method applied on queues

When a new view is requested, the lateral queue, with the files in low quality that belong to that view, is copied to the auxiliary and the other three are set to empty. While the files with the new view in high quality aren't in the central queue, that means the files in the auxiliary queue are sent to the Client. When the central queue is not empty and starts to receive the files requested, it starts to send them from the central queue again. After the transition to the central queue again, the lateral queues refills with future files of the surrounding views in low quality and proceeds with this behaviour until a new switching view operation is performed. The described mechanism is repeated for each switching view operation requested. The figure 4.6 illustrates this process in the buffering mechanism.

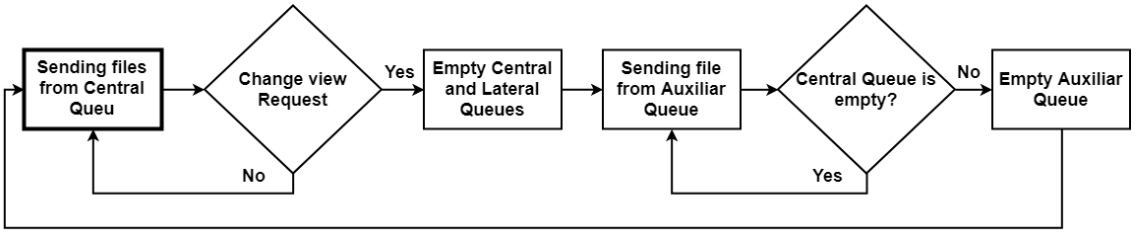

Figure 4.6: Flow chart Buffering Mechanism

4.7 New Test Prototype

As it was not possible to incorporate the Proxy in the Prototype in useful time, it was necessary to create a new test Prototype in order to validate the viability of the buffering mechanism. Below, it will be presented the architecture, the Client and the Server used in tests with the new test Prototype.

4.7.1 Architecture

The new test Prototype consists of three modules: Client, Proxy and Server. Client and Proxy communicate through a socket, which is responsible for sending the request for view and for forwarding the requested view in high quality to the Client. Proxy and Server communicate through four sockets. In a first phase, there is one socket that is responsible for requesting and sending the MPD file to the Proxy and the other three sockets are responsible to send the requested view in high quality and two surrounding views in low quality.

With the buffering mechanism and the MPD file sent by the Server, the Proxy is able to parse the files requested by the Client. The Server has the storage with the frame segments and the MPD file as well. Client has a five-button GUI that allows it to change the view and also contains a file name impression of the incoming files. The architecture of the Prototype is represented in the figure 4.7.

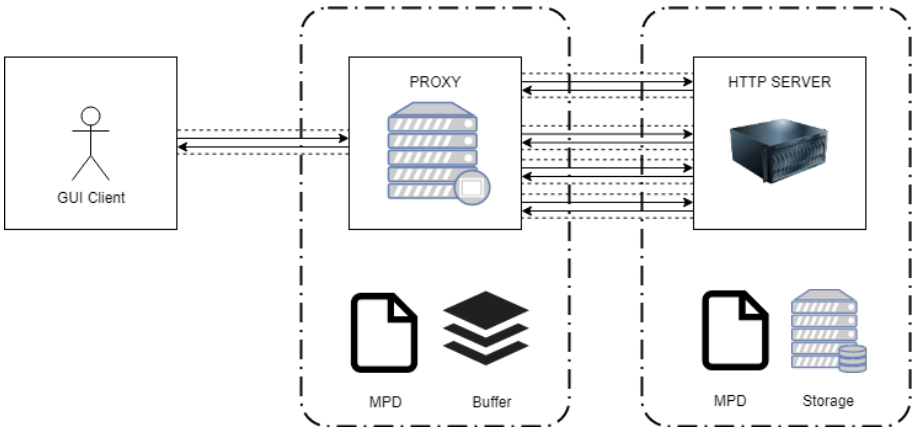

Figure 4.7: Architecture Diagram

4.7.2 Client

In the initial Prototype, the Client is divided in two modules: Player and MPEG-DASH Client. The Proxy wasn't incorporated in the initial Prototype and, for that reason a tool was developed to test and experiment the buffering mechanism. The Client developed in the new Prototype was created using as basis the Client from the Prototype above described in the attempt to recreate a manual switching operation.

Client has a GUI, which sends requests to Proxy through a socket following the HTTP protocol. This GUI consists of five buttons and each button allows a view transition and sends a view identification number to the Proxy.

In the GUI it's possible to see the files arriving with frame segments of the file's content and the time period since the request until the reception by the Client. This data will be important in test phases because it also gives information about the view transition time.

The buttons allow the user to request the following five views: Most Left view, Left view, Central view, Right view, Most Right view. Every click in a button represents a switch in view operation and sends a view ID to Proxy. Switching view operation and view ID are shown and explained in the section 4.7.4.

Finally, in the Client, it was incorporated a simulation of content reproduction, which consists in a counter that is shared with the other two modules providing a continuous flow of files, and a sleep command with the duration of 119 ms, which is the duration of the sent files. These two implementations allow a good approximation to the real reproduction of MPEG-DASH based content.

4.7.3 Server

The new Server created for the new test Prototype was based on the HTTP Server of the Prototype. Server communicates with the Proxy through four sockets, following the HTTP protocol. This Server has the MPD file and is responsible for sending it to Proxy, before the beginning of the experience. The content provided by storage server is sent using HTTP. This Server is composed by two modules: MPD, which is described in 4.5 and is responsible for describing all the available content, and the segments in the storage Server.

The Client starts the streaming session with the central view, but can select a view according to its preference after the beginning of the experience, using HTTP requests. During the experience, server is sending the subsequent segments allowing the flow of files in Client. After the MPD is sent, the Server has the main function of receiving the three URL of the views requested by Proxy and sends back the corresponding files. Server searches in storage for the requested URLs in order to find the correspondent requested files.

4.7.4 Switching Operation

Once, one of the buttons in GUI is selected, Client will sent a view ID to the Proxy. Every view is represented in the buttons, and each button has a view ID associated. The correspondence between view and view ID is represented in table 4.1.

Table 4.1: Correspondence between view and view ID

View	Most Left	Left	Central	Right	Most Right
View ID	4	2	1	3	5

When Proxy receives the view ID from Client, it will find in MPD files the URL correspondent to the requested view and the URL of surrounding views. Proxy does the search of the file using a filename as this: "ID_view quality_dashNR.m4s". Each segment of different views and different qualities has this type of filename. The part corresponding to ID is responsible for the type of view and quality. The range between [61-65] corresponds to lower quality and [91-95] to higher quality. In every range the first value corresponds to the Central view, the second to the Left view, the third to the Right view, the fourth to the Most Left view and finally the fifth to the Most Right view. The association between quality and range is in table 4.2. The part corresponding to NR in filename is responsible for the counter that synchronises the flow and every segment in time. The NR has the range of [1-1009] which corresponds to the number of segments that a view with a specific quality has. For example, "61_CenterPair verylow_dash1.m4s" corresponds to very low quality of central view in the first segment.

Table 4.2: Correspondence between Quality and URL ID interval

Quality	Very Low	Low	Medium	High Quality
URL ID interval	[61-65]	[71-75]	[81-85]	[91-95]

Therefore when the request is made the Proxy sends three URLs, one URL in the range of [91-95] which is the view requested in high quality, and the other two in the range [71-75] which represents the surrounding views in lower quality. The Server receives URLs and scans them in the storage, and sends them to Proxy. Every possible case for each every URL ID is described in the table 4.3.

Table 4.3: URL ID Cases for every central view and the surrounding views

URL ID Central view	91	92	93	94	95
URL ID Left view	72	74	71	-	73
URL ID Right view	73	71	75	72	-

4.8 List of assumptions

For the proper functioning of the new test Prototype some assumptions were made and will be presented next:

Each program has five views: Most Left, Left, Central, Right and Most Right, which one with four different quality modes.

MPEG-DASH multi-view Client consists of a main module with the MPEG-DASH player, a focus tracking module and the View Proxy module. For the development of this dissertation work, it is considered that the main module incorporates everything that already exists, namely the DASH player and the tracking of the attention focus.

The communication between the Client and the Proxy is synchronous and based on sockets. The MPD file used by the main module has a structure that allows to easily identify the segments

of each of the five views. The main module knows, at each moment, which view and quality it should ask in the initial connection to the server to obtain the MPD.

With the requests that the main module makes using the URLs obtained from the MPD presented in the Proxy and based in the request received from the main module, the Proxy is able to identify the two additional low quality views that it should request to the Server.

The requests that the Proxy makes to the server using the URLs obtained from the MPD-Server are established using three HTTP links with the server to order it from three different views. The request corresponding to each central view, is the request of the view with more quality, the other two requests are in low quality. These requests always use the same connection, which means the same socket. Responses are received and buffered through queues using a First in First out queuing algorithm.

To send the response to the parent module request, the Proxy must parse the URL of the request it receives and maintain an history record of only one request. If the new request has an URL that corresponds to a view segment that was already being viewed, it sends the data it received from the Server to the socket that corresponds to the higher-quality view. If the new request has an URL that is different, it must identify what view it is about. If it is the left view, it sends an auxiliary queue with the contents of the left queue and empties the other three queues and sends the low quality contents while the central queue is empty.

When the Proxy receives a new request from the main module, it always analyses that request to extract the URL and identify if it is a request for the same view. If the request is from the same view everything remains the same. If the request is different, then it has to ask the Server for this new view with higher quality on the socket that corresponds to the higher quality view and identify the views that it has to order with lower quality on the other two sockets.

4.9 Use Case

The Prototype is best aimed at a multi-view MPEG-DASH system that provides streaming. Client consists of two modules, a main module, which contains the multimedia player to display the multi-view content and has a component to monitor the user's focus. The other module is the Proxy, a secondary module, which is in charge of managing the transference of multimedia data between the Server and the Player. The figure 4.8 shows the functions of each module of the Prototype.

Client connects to Server and receives back a list of available content offered by the Server storage. Client selects its content of interest and sends the request to Server. The Server will return the MPD that corresponds to that content and Client will send a copy of the MPD to Proxy. The MPD is changed so that the existing URLs contain the IP address of the Proxy instead of the Server where the contents are stored.

So at this point there are two MPDs in Client, one in the Player and one in the Proxy, the only difference between them is the Server address. The MPD that resides in the Player is called

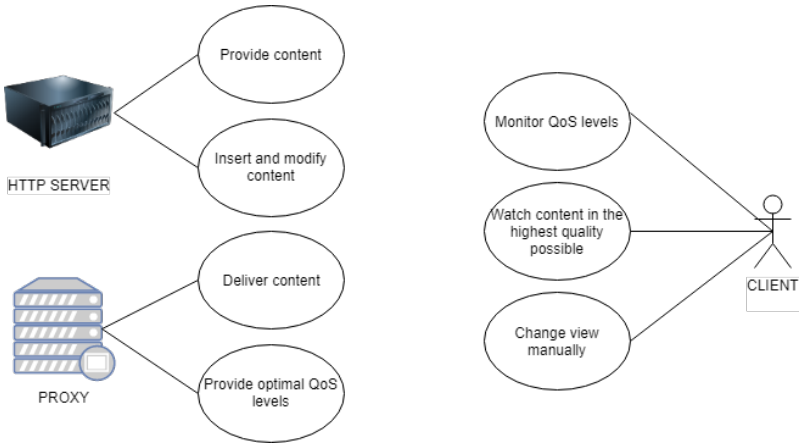

Figure 4.8: Use Case Diagram

MPD-Proxy because it contains Proxy addresses and what resides in Proxy is called MPD-Server because it contains Server addresses.

When the preview is initialised the play button is pressed and the current Player accesses the MPD-Proxy, locating the URLs of the segments corresponding to the central view with a medium quality. Sends a corresponding HTTP request that will be recited by Proxy.

The Proxy receives the URL and examines the MPD, interpreting it in order to order the view indicated by Client, in high quality. At this point, the IP address associated with the URL of the views will be adjusted to reflect the IP address of the Server where the content resides. The next step involves determining the URL of the surrounding views in order to send a low-quality version of those views.

In this way, the proxy will send three views, two of them in low quality and one of them in high quality that will be the view actually requested by the user. Only the high quality view is received in real time by Client, since it is the only view that is currently being played.

When there is a new request from the customer, a switch of the views is performed. The requested view will be one of the two that are being sent, which will be displayed in low quality while the server requests the high quality view. The process is repeated again and adjusted according to the new surrounding views.

Chapter 5

Tests and Experiments

The Test and Experiments section has the main goal of describing in detail the tests performed with the developed prototype and the respective results obtained in the simulations. The parameters selected to test are in section 3.3 . In this chapter, graphs will be presented and its respective analysis, and also a comparison of the results obtained in the different experiences. This chapter provides several information regarding the mechanism developed for this dissertation.

5.1 Approach

In order to understand the flexibility of the buffering mechanism two simulations were performed. Simulations were tested in two different network conditions. In the first simulation, the prototype without any buffer mechanism was tested and in the second simulation tested the prototype with the buffer mechanism. The network conditions used in simulation and measured in Proxy are represented in table 5.1.

Table 5.1: Network Conditions

Network	Type 1	Type 2
Receiving (Mbyte/s)	2.5	6,5
Sending (Mbyte/s)	1.4	2.3

Different network conditions provide distinct data which allow us to understand the mechanism's flexibility. It was essential to understand how these conditions can limit the buffer and its possible application to real environments.

In Simulation A(5.2) and B (5.3) the following data was collected: the view transition operation duration and the time period between the request from Proxy and the reception of packages from Server. The data collected in Simulation A is important to find differences between using or not using the buffer mechanism developed during this work. Together with Simulation B it was possible to evaluate the performance of the mechanism developed.

5.2 Simulation A: Prototype without the buffer mechanism

5.2.1 Test 1: View transition performance according to network conditions 1

As previously mentioned, the experiment only starts after the client receives the initial MP4 file. After some experiments, it was noticed that this process average time was 146.66 seconds. Additionally, it is important to mention that in order to understand the performance of the developed algorithm, all the switching operations processing times were examined. The considered to be the most relevant values are shown in table 5.2. As it is possible to notice the worst case scenario switching view duration is around 0.127 seconds. Furthermore, the best case scenario is sharply 0.965 seconds long, while the operation average duration is 0.111 seconds.

Table 5.2: Test 1 View switching Average Latency

View	Lateral View	Average Latency(s)
Central	Left	0.111025
	Right	0.120675
Left	Most Left	0.108875
	Central	0.111355
Right	Central	0.096445
	Most Right	0.110925
Most Left	Left	0.126690
Most Right	Right	0.112500

Figure 5.1 depicts a simulation of a real switching view experience. As it can be noticed, the Proxy struggles when a new view is requested by a Client. This is noticeable in the chart peaks which simply represent a switching view operation. It should be mentioned that the chart is normalised regarding the total duration of the considered test.

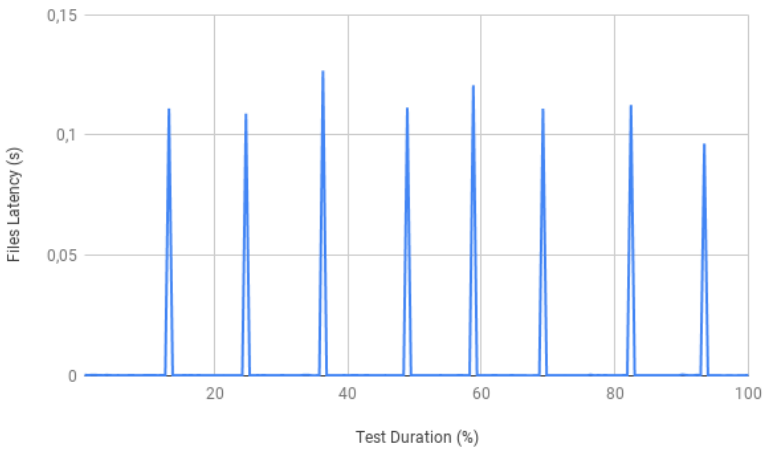

Figure 5.1: Test 1 Flow of files received by Client

Focusing in the expected reception process duration values and the observed in the tests, it was obtained the high quality files reception time and consequently its average value. This allowed to understand that the biggest switching view operation sub delay occurred in the Proxy-Server connection. Table 5.3 shows the total and sub delay perspectives of the obtained values.

Table 5.3: Test 1 Sub Delay Average

	Client- Proxy	Proxy-Server	Total
Average Sub Delay	0.00027	0.11932	0.11959

5.2.2 Test 2: View transition performance according to network conditions 2

As in the previous simulation test, the data collected corresponded to the duration of the reception of the initial file. After some experiments, it was concluded that this process average time was 136.47 seconds. As well, it is important to consider that in order to test the performance of the developed algorithm, all the switching operations processing times were collected. The considered to be the most relevant values are shown in table 5.4. As it is possible to verify, the worst case scenario switching view duration is around 0.088 seconds. Additionally, the best case scenario is sharply 0.058 seconds long, while the operation average duration is 0.068 seconds.

Table 5.4: Test 2 View switching Average Latency

View	Lateral View	Average Latency(s)
Central	Left	0.058485
	Right	0.077105
Left	Most Left	0.085470
	Central	0.064795
Right	Central	0.07469
	Most Right	0.063625
Most Left	Left	0.088005
Most Right	Right	0.061975

Figure 5.2 depicts a simulation of a real switching view experience. As it can be noticed, the Proxy has the same behaviour when a new view is requested by a Client. Once more, this is depicted in the chart peaks which represent a switching view operation and the chart is normalised regarding the total duration of the considered test.

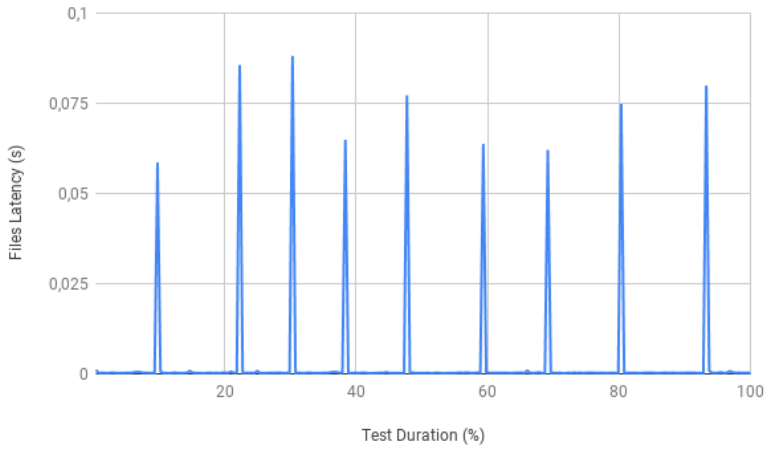

Figure 5.2: Test 2 Flow of files received by Client

One more time, focusing in the expected reception process duration and the observed in the tests, it was obtained the high quality files reception time and consequently its average value. This allowed to observe that the biggest switching view operation sub delay occurred between the Proxy and the Server. Table 5.5 shows the total and sub delay average values obtained. Hence, it is noticeable that the algorithm has a similar behaviour. However, the observed peaks are in a lower scale than the ones noticed on the previous test.

Table 5.5: Test 2 Sub Delay Average

	Client- Proxy	Proxy-Server	Total
Average Sub Delay	0.00027	0.06813	0.06834

5.3 Simulation B: Prototype with buffer mechanism

5.3.1 Test 1: View transition performance according to network conditions 1

As in the other tests, the same initial condition were respected. With the collected data it was observed that the initial process average time was 137.04 seconds. Furthermore, it is important to obtain data of all the switching view operations processing times. This will allow to test the performance of the developed algorithm. The most relevant values collected are shown in table 5.6. As it is possible to notice the worst case scenario switching view is around 0.00039 seconds. As well to, the best case scenario is 0.00028 seconds long, while the operation average duration is 0.00033 seconds.

Table 5.6: Test 1 View switching Average Latency

View	Lateral View	Average Latency(s)
Central	Left	0.00028
	Right	0.00036
Left	Most Left	0.00031
	Central	0.00039
Right	Central	0.00033
	Most Right	0.00031
Most Left	Left	0.00043
Most Right	Right	0.00032

Figure 5.3 depicts an example of a simulation of real view switching operation. As it can be noticed, the Proxy has a similar behaviour as the one observed in previous tests. However, when a new view is requested by a Client even lower operation duration values are observed when compared to the obtained results in the Prototype without the buffer. Then, it is noticed a flow decrease that is related with low quality files being sent, stabilising when finally the Proxy is able to send the high quality ones. It should be mentioned that the chart is normalised regarding the total duration of the considered test.

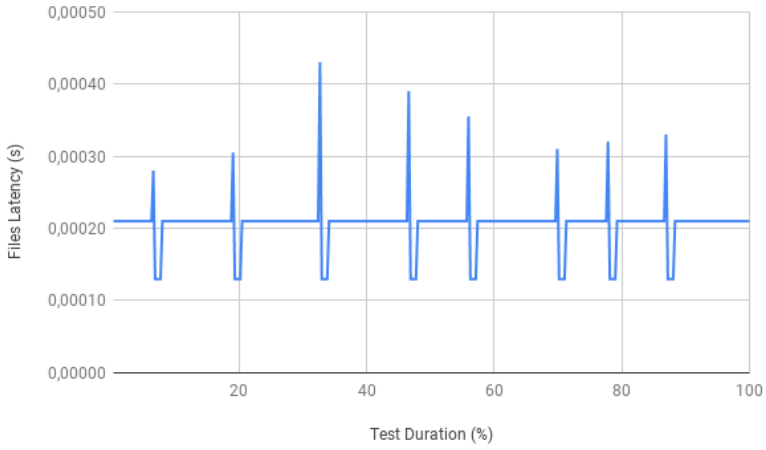

Figure 5.3: Test 1 Flow of files received by Client

Other informations were collected in order to better understand the mechanism performance. It was realised that before the stabilisation achievement, the Proxy averagely sends five low quality files and this process has the duration average of 0.00101 seconds. The reception duration of the low quality file was also collected and its value is 0.00013 seconds. On the other hand, the high quality ones have a data exchanging process of 0.00021 seconds.

5.3.2 Test 2: View transition performance according to network conditions 2

As in the other tests, the same initial condition were respected. After collection of data, it was observed that the initial process average time was 131.63 seconds. Furthermore, it is once more important to obtain data of all the switching view operations processing times. This will allow to test the performance of the developed algorithm. The most relevant values collected are shown in table 5.7. As it is possible to notice the worst case scenario switching view is around 0.00039 seconds. As well to, the best case scenario is 0.00028 seconds long, while the operation average duration is 0.00030 seconds.

Table 5.7: Test 2 View switching Average Latency

View	Lateral View	Average Latency(s)
Central	Left	0.00034
	Right	0.00029
Left	Most Left	0.00028
	Central	0.00031
Right	Central	0.00034
	Most Right	0.00029
Most Left	Left	0.00036
Most Right	Right	0.00027

Figure 5.4 depicts an example of a simulation of real view switching operation. As it can be noticed, the Proxy has a similar behaviour as the one observed in previous test. However, when a new view is requested by a Client even lower operation duration values are observed when comparing to the obtained results in the Prototype without the buffer. Then, it is noticeable a following flow decrease that is related with low quality files being sent, stabilising when finally the Proxy is able to send the high quality ones. It should be mentioned that the chart is normalised regarding the total duration of the considered test.

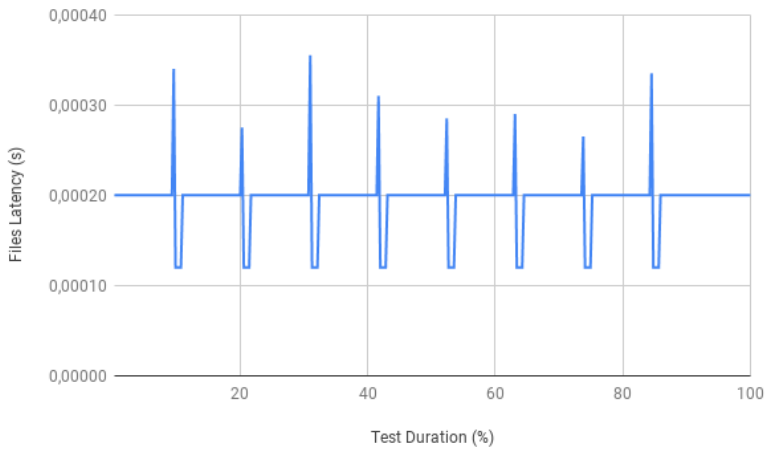

Figure 5.4: Test 2 Flow of files received by Client

Other information were collected in order to understand the better performance of the mechanism. It was realised that before the stabilisation achievement, the Proxy averagely sends five low quality files and this process has the duration average of 0,00101 seconds. The reception duration of the low quality file was also collected and its value is 0.00012 seconds. On the other hand, the high quality ones have a data exchanging process of 0.00020 seconds.

5.4 Comparison

In table 5.8 it can be observed the obtained values in the different performed tests. With the network condition 1 we can observe a 99.7% decrease on the switching view latency. Also, in the network condition 2 we can observe a 99.5% decrease on the the same operation. Hence, the newly created buffer mechanism allows for an improved user's experience when compared to the one that was initially implemented.

Table 5.8: Comparison View switching Average Latency

	Average Latency (s)		Decrease
	No Buffer mechanism	Buffer mechanism	
Network type 1	0.111	0.00033	99,7%
Network type 2	0.068	0.00030	99,5%

Figure 5.5 represents an example of a simulation of two real time switching view experiences in different network environments. This allows to compare the two type of network conditions and it can be observed a small difference on the latency time, but is not sufficiently significant.

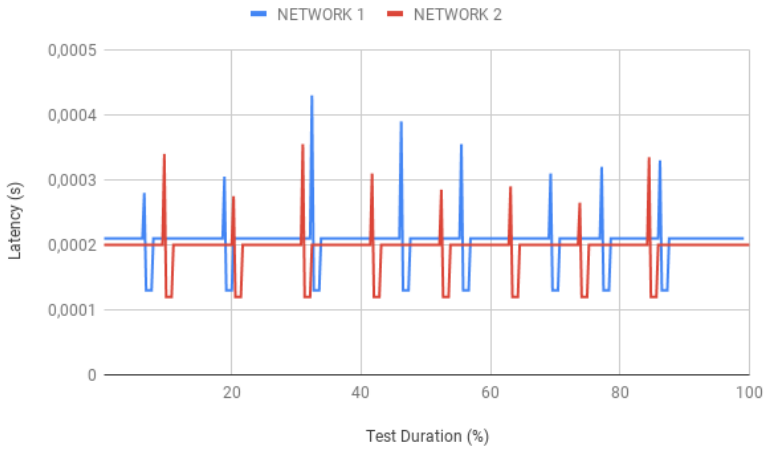

Figure 5.5: Comparison flow of files received by Client

Chapter 6

Conclusions and Future Work

The chapter 6 is an overview of the work done and an analysis of the objectives proposed and completed in this dissertation. It provides a reflection upon the results obtained in the tests and experiments executed with the developed mechanism. Finally, some suggestions to be considered in order to improve the buffer mechanism and the Prototype in future works are made.

6.1 Results and Conclusions

The main objective of this dissertation was to offer a better QoE in a multi-view streaming experience with lower costs. For that, an analyses was performed to identify major limitations and difficulties of the previous Prototype. In order to overcome those limitations, it was designed a buffer mechanism. The developed mechanism is flexible and can be applied in different network environments. It was designed for multi-view immersive experiences and was tested in a five different view environment.

This algorithm provides a switching view operation for surrounding views without compromising QoS, and consequently QoE, of a multi-view experience. It is important to note that the algorithm was conceived to avoid losing files during the switching view operation. The tests performed suggest that this algorithm reduces the duration of a switching view operation. Additional tests were performed to collect data with information about received files flow and also about the latency during view switching operation.

To notice that, despite a usually smooth experience provided by the implemented Proxy, the algorithm struggles when some events occur. Under network congestion scenarios was observed that it could behave unexpectedly. However, because this type of network environment was not a priority scenario in the objectives of this study, these problems did not affect the obtained results.

The high peaks observed in the graphs that compare the flows of received files show a decrease in the latency of view switching operation. The duration of the view transition operation with the implemented mechanism is approximately the time needed to send a file. The data collected confirms that the buffering mechanism achieves the proposed objectives.

Average switching latency with Proxy in network conditions 1 decrease 99.% and in network conditions 2 decrease 99.5%, which reflects a minimal switching duration. The view switching latency problem is overcome and the switching operation is almost immediate. The duration of switching operation with the buffer mechanism corresponds sharply to the time needed to receive a low quality file of that view and receives the high quality files only 0.00101 seconds after. This switching to high quality has the approximated reception duration of any file received from Proxy and that provides a continuous flow of received files.

This system supports switching view operations without negative interference in the experience. The results obtained suggest that the buffer mechanism supports experiences with this type of files without compromising the QoS. It is important to note that no files are lost during the switching experience and that provides a better performance. Because of that, the algorithm is suitable to be applied in a real environment scenario as it was initially proposed.

Finally, the mechanism developed for switching views presented in this dissertation can be applied in any common streaming service using the On Demand profile. This profile offers advantages in worst network conditions. The Buffer mechanism can be adapted to support an higher number of views and different quality modes, and good performance results are expected without compromising the QoS/QoE. It is important to note that the initial conditions proposed in this dissertation must be followed to obtain the presented performance and MPD has to be modified if any view or quality is added to the storage.

6.2 Future Work

For future work, the algorithm should be improved in order to perfectly handle high congestion network environments. The next step to take into account is the Proxy incorporation in the Prototype environment. These buffer mechanisms have good results dealing with the same files used in the Prototype and also in reducing the latency during switching view operation. To incorporate the Proxy it will be necessary to modify the addresses to Server in Client and the URLs presented in MPD need to be modified to the Proxy address. This Proxy has a limitation, which is the fact that it has only been tested in an On Demand profile. It needs an improvement to be able to deal with other profiles in order to become more flexible and complete.

The Prototype uses the codec AVC/H.264 and that can be a limitation. It can't support streams coded with newest versions. HEVC looks like a good solution because provides advantages to the system. The research for this implementation must be considered in order to take advantages of HEVC. HEVC offers advantages in video compression, providing the same quality of delivery with a lower bit rate. This codec provides high quality live streaming quickly and efficiently answering the growing demand for high quality content.

Another suggestion for future research is to apply the prediction of human behaviour techniques using Deep Learning. This method requires some AI in order to predict human action. Recently, tests with deep learning algorithms created to predict human behaviour revealed that

this algorithm is able to predict 43% of human actions, correctly.[65] Adding this prediction techniques is an interesting improvement to the experience and to the functionality and flexibility of the buffering mechanism. It is necessary to study this method and the possibility of its integration into the existing Prototype and associated developed mechanism. This method will be very useful predicting the user's attention focus. The buffer could cooperate with this mechanism in order to prepare the view to be presented in advance.

References

[1] G. Sullivan and T. Wiegand, "Video compression - from concepts to the h.264/avc standard," *Proceedings of the IEEE*, vol. 93, pp. 18 – 31, 02 2005.

[2] G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, 12 2012.

[3] "Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications,joint ad hoc group for digital representations of light/sound fields for immersive media applications," 2016.

[4] C. J. Kale and T. J. Socolofsky, "TCP/IP tutorial." RFC 1180, Jan. 1991.

[5] S. H. J. S. Kyuheon Kim, Kyungmo Park, "Draft of white paper on mpeg media transport (mmt)," *IEEE*, 2015.

[6] A. Inc., "Understanding the http live streaming architecture." https://developer.apple.com/documentation/http_live_streaming, visitado em Fevereiro 2018.

[7] C. Mueller, "Mpeg-dash (dynamic adaptive streaming over http)." https://bitmovin.com/dynamic-adaptive-streaming-http-mpeg-dash/, visitado em Fevereiro 2015.

[8] T. Costa, M. Andrade, and P. Viana, "Predictive multi-view content buffering applied to interactive streaming system," *Electronics Letters*, 05 2019.

[9] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, pp. 62–67, 2011.

[10] Statista, "Forecast for the number of active virtual reality users worldwide from 2014 to 2018 (in millions)." https://www.statista.com/statistics/426469/active-virtual-reality-users-worldwide/, visit in Fevereiro 2018.

[11] IDC, "Augmented reality and virtual reality headsets poised for significant growth, according to idc." https://www.idc.com/getdoc.jsp?containerId=prUS44966319, visit in May 2019.

[12] Forbes, "Are vr and ar the future of live events?." https://www.forbes.com/sites/solrogers/2018/11/26/are-vr-and-ar-the-future-of-live-events/, visit in Fevereiro 2018.

[13] Sapo, "Os live em 360º do facebook agora podem ser vistos em 4k e em realidade virtual." http://bit.do/360-live-4k-streaming, visit in Fevereiro 2017.

[14] Qualcomm, "Immersive experiences." https://www.qualcomm.com/invention/cognitive-technologies/immersive-experiences, visit in May 2019.

[15] Lifewire, "What is an immersive experience?." https://www.lifewire.com/what-is-an-immersive-experience-4588346, visit in May 2019.

[16] G. S. Jens-Rainer Ohm, "Mpeg-4 part 10." https://mpeg.chiariglione.org/standards/mpeg-4/advanced-video-coding, 2005.

[17] I. Richardson, "White paper: An overview of h.264 advanced video coding." http://www.videosurveillance.co.in/H.264.pdf, visitado em Fevereiro 2007.

[18] P. Merkle, K. Müller, and T. Wiegand, "3d video: acquisition, coding, and display," *IEEE Transactions on Consumer Electronics*, vol. 56, pp. 946–950, May 2010.

[19] J.-R. Ohm and G. Sullivan, "Mpeg-h part 2." https://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding, 2011.

[20] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Müller, H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3d video coding using advanced prediction, depth modeling, and encoder control methods," in *2012 Picture Coding Symposium*, pp. 1–4, May 2012.

[21] Vcodex, "Hevc: An introduction to high efficiency coding." https://www.vcodex.com/hevc-an-introduction-to-high-efficiency-coding/, visitado em Fevereiro 2018.

[22] G. Developers, "Degrees of freedom." https://developers.google.com/vr/discover/degrees-of-freedomf, visitado em Fevereiro 2018.

[23] I. B. Thomas Stockhammer, "Mpeg -i architectures," 2018.

[24] K. Farr, "Apple joins av1 codec consortium. what does it mean for you?." https://bitmovin.com/apple-joins-av1-codec-consortium/, visitado em Fevereiro 2018.

[25] Bitmovin, "Abitmovin e av1." https://bitmovin.com/av1/, visitado em Fevereiro 2018.

[26] C. Feldmann, "Cool new video tools: Five encoding advancements coming in av1." https://bitmovin.com/cool-new-video-tools-five-encoding-advancements-coming-av1/, visitado em Fevereiro 2018.

[27] AOMedia, "A shared vision for the future of video streaming." https://aomedia.org/news/a-shared-vision-for-the-future-of-video-streaming/, visitado em Fevereiro 2018.

[28] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," *Internet Engineering Task Force, RFC*, 07 2003.

[29] D. J. D. Touch, "Transport Options for UDP," Internet-Draft draft-ietf-tsvwg-udp-options-07, Internet Engineering Task Force, Mar. 2019. Work in Progress.

[30] "User Datagram Protocol." RFC 768, Aug. 1980.

[31] L. Eggert, G. Fairhurst, and G. Shepherd, "UDP Usage Guidelines." RFC 8085, Mar. 2017.

[32] 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, pp. 469–492, Firstquarter 2015.

[33] "Use cases for mpeg media transport (mmt)," *IEEE*, 2013.

[34] R. Pantos, "HTTP Live Streaming 2nd Edition," Internet-Draft draft-pantos-hls-rfc8216bis-04, Internet Engineering Task Force, Mar. 2019. Work in Progress.

[35] R. Pantos and W. May, "HTTP Live Streaming." RFC 8216, Aug. 2017.

[36] D. Yun and K. Chung, "Dash-based multi-view video streaming system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 1974–1980, Aug 2018.

[37] E. Thomas, M. van Deventer, T. Stockhammer, A. Begen, and J. Famaey, "Enhancing mpeg dash performance via server and network assistance," *SMPTE Motion Imaging Journal*, vol. 126, pp. 22–27, 01 2017.

[38] O. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Yong Lim, "Mpeg dash srd: spatial relationship description," pp. 1–8, 05 2016.

[39] A. Hamza and M. Hefeeda, "A dash-based free viewpoint video streaming system," p. 55, 03 2014.

[40] I. Sodagar, "White paper on mpeg-dash's new features," *Communication*, 2017.

[41] M. Oskar van Deventer, E. Thomas, F. Mazé, and F. Denoual, "White paper on mpeg dash part 1 amd 2 spatial relationship descriptor (srd)," 08 2016.

[42] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," in *2017 IEEE Virtual Reality (VR)*, pp. 423–424, March 2017.

[43] S. Y. Lim, J. M. Seok, J. Seo, and T. G. Kim, "Tiled panoramic video transmission system based on mpeg-dash," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 719–721, Oct 2015.

[44] R. Dubin, O. Hadar, and A. Dvir, "The effect of client buffer and mbr consideration on dash adaptation logic," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2178–2183, April 2013.

[45] T. Group, "This is eye tracking." https://www.tobii.com/group/about/this-is-eye-tracking/, visitado em Fevereiro 2019.

[46] A. Barreto, "Eye tracking como método de investigação aplicado às ciências da comunicação," *Revista Comunicando*, 12 2012.

[47] R. Lupu and F. Ungureanu, "A survey of eye tracking methods and applications," *Buletinul Institutului Politehnic din Iaşi. Secţia Automatică şi Calculatoare*, vol. 3, 01 2013.

[48] T. Group, "Education." https://www.tobiipro.com/fields-of-use/education/, visit in Fevereiro 2018.

[49] T. Group, "Psychology and neuroscience." https://www.tobiipro.com/fields-of-use/psychology-and-neuroscience/, visit in Fevereiro 2018.

[50] R. Schapire, "Machine learning algorithms for classification. technical report, princeton university." https://www.cs.princeton.edu/~schapire/talks/picasso-minicourse.pdf, visitado em Fevereiro 2013.

[51] J. Brownlee, "A tour of machine learning algorithms." https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/, visitado em Fevereiro 2013.

[52] J. L. Castellanos, M. F. Gomez, and K. D. Adams, "Using machine learning based on eye gaze to predict targets: An exploratory study," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, Nov 2017.

[53] D. Saha, M. Ferdoushi, M. Tanvir Emrose, S. Das, S. M Mehedi Hasan, A. Intisar Khan, and C. Shahnaz, "Deep learning-based eye gaze controlled robotic car," pp. 1–6, 12 2018.

[54] Y. Yin, C. Juan, J. Chakraborty, and M. P. McGuire, "Classification of eye tracking data using a convolutional neural network," pp. 530–535, 12 2018.

[55] F. Koochaki and L. Najafizadeh, "Predicting intention through eye gaze patterns," pp. 1–4, 10 2018.

[56] "Python." https://www.python.org/, 2019.

[57] "Ubuntu." https://www.ubuntu.com/, 2019.

[58] "Oracle vm virtualbox." https://www.virtualbox.org/, 2019.

[59] "Overleaf, online latex editor." https://www.overleaf.com/, 2019.

[60] J. D. F. S. Costa, "Automatic adaptation of views in 3d applications," Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2016.

[61] "Mp4client." https://gpac.wp.imt.fr/player/, 2019.

[62] "Mp4box." https://gpac.wp.imt.fr/mp4box/, 2019.

[63] "Gpac and dash." https://gpac.wp.imt.fr/player/features/dash/, 2019.

[64] "Expat xml parser." https://libexpat.github.io/, 2019.

[65] A. UJ, "Predicting human behaviour with deep learning." https://www.analyticsinsight.net/predicting-human-behaviour-with-deep-learning/, 2019.