# Universitat Politècnica de València

## Departamento de Comunicación Audiovisual, Documentación e Historia del Arte


# Tesis Doctoral

Design of an architecture and communication protocol for video transmission and videoconference

## Presented by:


José Miguel Jiménez Herranz


## Supervised by:


Jaime Lloret Mauri


**Valencia, July 2018**

# Abstract

This doctoral thesis faces the problem of video transmission, in television transmission systems and video on demand in real time, and video conferencing, in IP networks, making a special emphasis when the connection is established between mobile devices.

In the first part of the work, video transmission in IP networks is introduced, and a special emphasis is made on videoconference, afterwards presenting the state of the art in both areas. We address several issues related to video compression, the quality of the video received, the Quality of Experience (QoE), systems for improving the QoE for videoconferencing and also technologies, such as Software Defined Networking (SDN), which allow us to dynamically improve the configuration of the network devices to improve the QoE perceived by the end users.

Later, we come to describe the design of the architecture and protocol for the transmission of video in IP networks, with different objectives. We propose an architecture and a protocol to improve the QoE of the end users in the transmission of video in Internet Protocol Television (IPTV). We also do it for the transmission of video in Heterogeneous Networks using HTML5. Finally, we propose an algorithm and protocol to improve QoE of video transmission in networks used for environmental monitoring, fundamentated mainly on a study substantiated on coding based on the predominant color of the images of the video.

In the next chapter, we propose a new architecture and a protocol for improving the End-to-End Quality of Experience (E2E QoE) of the users connected to a video conference. We define the process of the system,

the finite state machine and the algorithm necessary for its correct operation.

Finally, we present the different experimental tests that have been carried out to study and determine the best coding systems, and their behavior when using technologies such as SDN in networks of mobile devices, in Heterogeneous Networks using HTML5 and in videoconference.

# Resumen


Esta tesis doctoral aborda el problema de la transmisión de vídeo, en sistemas de transmisión de televisión y vídeo bajo demanda en tiempo real y de videoconferencia, a través de redes IP, haciendo un énfasis especial cuando la conexión se establece entre dispositivos móviles.

En la primera parte del trabajo se introduce la transmisión de video en redes IP y se hace especial incapie en la videoconferencia, pasando posteriormente a presentar el estado del arte en ambos ámbitos. Tratamos y estudiamos cuestiones relacionadas con la compresión de vídeo, la calidad del vídeo recibido, la calidad de experiencia (QoE), sistemas para la mejora de la QoE para la videoconferencia y también tecnologías, como Software Defined Networking (SDN), que nos permiten mejorar dinámicamente la configuración de los dispositivos de red para mejorar la QoE percibida por los usuarios finales.

Posteriormente, pasamos a describir el diseño de la arquitectura y el protocolo para la transmisión de vídeo en redes IP, con distintos objetivos. Proponemos una arquitectura y un protocolo para mejorar la QoE de los usuarios en la transmisión de vídeo en Internet Protocol Television (IPTV). También lo hacemos para la transmisión de vídeo en Redes Heterogéneas usando HTML5. Por último proponemos un algoritmo y protocolo para mejorar QoE de la transmisión de vídeo en redes empleadas para la monitorización ambiental, basándonos principalmente en un estudio fundamentado en la codificación en función del color predominante de las imágenes del vídeo.

En el siguiente capítulo pasamos a proponer una nueva arquitectura y un protocolo para la mejora de la calidad de experiencia de extremo a

extremo (E2E QoE) de los usuarios conectados a una videoconferencia. Definimos el proceso del sistema, la máquina de estados finita y el algoritmo necesario para su correcto funcionamiento.

Por último, presentamos las distintas pruebas experimentales que se han realizado para estudiar y determinar los mejores sistemas de codificación, y su comportamiento al utilizar tecnologías como SDN en redes de dispositivos móviles, en Redes Heterogéneas usando HTML5 y en videoconferencia.

# Resum

Aquesta tesi doctoral aborda el problema de la transmissió de vídeo, en sistemes de transmissió de televisió i vídeo sota demanda en temps real, i de videoconferència, a través de xarxes IP, fent un èmfasi especial quan la connexió s'estableix entre dispositius mòbils.

En la primera part del treball s'introdueix la transmissió de vídeo en xarxes IP i es fa especial incapie en la videoconferència, passant posteriorment a presentar a l'estat de l'art en tots dos àmbits. Tractem i estudiem qüestions relacionades amb la compressió de vídeo, la qualitat del vídeo rebut, la qualitat d'experiència (QoE), sistemes per a la millora de la QoE per a la videoconferència i també tecnologies, com a Software Defined Networking (SDN), que ens permeten millorar dinàmicament la configuració dels dispositius de xarxa per a millorar la QoE percebuda pels usuaris finals.

Posteriorment, passem a descriure el disseny de l'arquitectura i el protocol per a la transmissió de vídeo en xarxes IP, amb diferents objectius. Proposem una arquitectura i un protocol per a millorar la QoE dels usuaris en la transmissió de vídeo en Internet Protocol Television (IPTV). També ho fem per a la transmissió de vídeo en Xarxes Heterogènies usant HTML5. Finalment proposem un algoritme i un protocol per a millorar QoE de la transmissió de vídeo en xarxes emprades per al monitoratge ambiental, basant-nos principalment en un estudi fonamentat en la codificació en funció del color predominant del vídeo.

En el següent capítol passem a proposar una nova arquitectura i un protocol per a la millora de la qualitat d'experiència d'extrem a extrem (I2I QoE) dels usuaris connectats a una videoconferència. Definim el procés del

sistema, la màquina d'estats finita i l'algoritme necessari per al seu correcte funcionament.

Finalment presentem les diferents proves experimentals que s'han realitzat per a estudiar i determinar els millors sistemes de codificació, i el seu comportament en utilitzar tecnologies com SDN en xarxes de dispositius mòbils, en Xarxes Heterogènies usant HTML5 i en videoconferència.

# Acknowledgements

Para completar esta tesis he realizado un viaje a lo largo de estos últimos años, en el que he ampliado tanto mi aprendizaje profesional como personal, vinculado a la investigación científica. Ha sido un trabajo en el que siempre he encontrado ayuda y apoyo incondicional de mi director de tesis, Ph. D. Jaime Lloret Mauri, con el que he disfrutado de una gran cantidad de tiempo proponiendo y ajustando las ideas y los experientos que se han llevado a cabo y finalizado de forma positiva, obteniendo como resultado la presentación de la tesis. Su ayuda ha sido imprescindible, sobre todo en aquellas situaciones que presentaban una gran dificultad para la obtención de resultados, momentos en los que la moral y los animos sufren una gran bajada, y él siempre demostraba un entusiasmo y fortaleza que transmitía constantemente. También quiero agradecer su esfuerzo por la revisión de cada uno de los diferentes trabajos desarrollados y la revisión final de la tesis.

A lo largo de este viaje, he compartido también mis dudas y descubrimientos con compañeros, que en todo momento me han prestado una generosa colaboración, tanto en la preparación de material como realización de test que han permitido obtener los resultados que presento en esta tesis. Ha sido muy valiosa para mí la ayuda proporcionada por Juan Ramón Díaz Santos, Sandra Sendra Compte, Lorena Parra Boronat, Oscar Romero Martinez, Albert Rego Mañez, Alejandro Canovas Solves, Laura Garcia Garcia y Miran Taha.

Quiero destacar el consejo, asesoramiento y paciencia de mi amigo Juan Ramón. Siempre he tenido en cuenta sus buenos consejos, que me han proporcionado puntos de vista diferentes a la hora de tomar decisiones para solucionar los problemas que se han presentado. Ha sido un gran compañero de trabajo a lo largo de muchos años, realmente el compañero y amigo que todos deseamos tener. Siempre he encontrado en él a la persona comprometida y afable que presta su ayuda de forma desinteresada, siendo un ejemplo de humildad y buen hacer. Gracias Juan por permitirme ser tu amigo.

También quiero agradecer la ayuda proporcionada por Vicente Ferran de Yranzo, para mí un hermano con el que no comparto apellidos, tanto de forma moral como material.

No puedo olvidar en este apartado de agradecimientos a Pascal Lorenz, de la Universidad de Haute Alsace, que me acogió durante mi estancia en su laboratorio de investigación.

Por último, quiero agradecer a Teresa mi esposa la paciencia, el cariño, el apoyo y la ayuda que siempre ha mostrado en momentos de adversidad, y que me ha permitido superar los retos que se han presentado.

Gracias a todos.

# Abbreviations

| | |
|---|---|
| **5G** | Fifth Generation Wireless Systems |
| **AP** | Access Point |
| **ARTS** | Advanced Real Time Streaming |
| **ASICs** | Application Specific Integrated Circuits |
| **AT&T** | American Telephone and Telegraph |
| **AVC** | Advanced Video Coding |
| **BRV** | Bit Rate |
| **CCITT** | International Telegraph and Telephone Consultative Committee |
| **CDNs** | Content Delivery Networks |
| **DCT** | Discrete Cosine Transform |
| **DHCP** | Dynamic Host Configuration Protocol |
| **DSIS** | Double Stimulus Impairment Scale |
| **E2E EoS** | End-to-End Quality of Experience |
| **E2E QoS** | End-to-End Quality of Service |
| **EPO** | European Patent Office |
| **fps** | frames per second |
| **GOP** | Group of Pictures |
| **HDLC** | High-Level Data Link Control |
| **HTML** | HyperText Markup Language |
| **HVQ** | Hierarchical Vectors Quantification |
| **IEC** | International Electrotechnical Commission |

**IETF** Internet Engineering Task Force

**IMS** IP Multimedia Subsystem

**IPTV** Internet Protocol Television

**IPv4** Internet Protocol version 4

**IPv6** Internet Protocol version 6

**ISDN** Integrated Services Digital Network

**ISO** International Organization for Standardization

**ISP** Internet Service Provider

**ITU** International Telecommunication Union

**ITU-T** ITU Telecommunication Standardization Sector

**JND** Just Noticeable Difference

**LAN** Local Area Network

**MOS** Mean Opinion Score

**MPEG** Moving Picture Experts Group

**MPLS** Multiprotocol Label Switching

**NFV** Network Function Virtualization

**NGN** Next Generation Networks

**NIC** Network Interface Card

**NOS** Network Operating System

**NSA** National Security Agency

**NTSC** National Television Standards Committee

**OSHI** Open Source Hybrid IP

**PAL** Phase Alternating Line

**PBN** Packet-Based networks

**PCM** Pulse Code Modulation

**PSNR** Peak Signal-to-Noise Ratio

**P2P** Peer-to-Peer

**PVST +** Per Vlan Spanning Tree Plus

**QoE** Quality of Experience

**QoS** Quality of Service

RAE Real Academia Española

**RFC** Request for Comments

**RSSI** Radio Signal Strength Indicator

**RTP** Real-Time Transport Protocol

**RTSP** Real-Time Streaming Protocol

**RTT** Round Trip Time

**SECAM** Séquentiel Couleur à Mémoire

**SDN** Software Defined Networks

**SSRC** Synchronization Source

**SVC** Scalable Video Coding

**TCP** Transmission Control Protocol

**UDP** User Datagram Protocol

**VCEG** Video Coding Experts Group

**SOA** Service Oriented Architecture

**VM** Virtual Machine

**VLC** VideoLAN Client

**VQEG** Video Quality Experts Group

**VQM** Video Quality Monitor

**VoD** Video-on-Demand

**WiMAX** Wireless Interoperability for Microwave Access

**Wi-Fi** Wireless Fidelity

**WLAN** Wireless Local Area Network

# Table of Contents

# List of figures

# List of tables

# Chapter 1. Introduction

# 1.1 Introduction

Real-time video connectivity is becoming a key feature in a wide range of services, including communications, corporate collaboration, social networking, gaming and entertainment. The number of these services, as well as the corresponding demand for device and network resources, is increasing at an exponential rate.

The challenges of efficient resource utilization and End-to-End Quality of Experience (E2E QoE) management are particularly critical in resource-constrained environments, like, mobile cellular networks. - Besides, mobile networks have great disadvantage, constantly fluctuate in bandwidth, delay, jitter, and packet loss.

Nowadays, the deployment of multimedia services in wired, mobile and wireless networks, with sufficient Quality of Experience (QoE) for end user, is a problem not yet solved at a practical level. Recently, it has been recognized that wireless and mobile networks are slowing down the deployment of this type of services. Current systems have great limitations due to their rigidity, which is a result of static configurations based principally on static commands or scripts, and is also unable to respond to these problems and offer multimedia services with the required quality. Automation of resources' supply lessens to such an extent that efficiency decreases. In addition, virtualization and cloud technologies are radically changing the traffic patterns within the data centers [1]. That is mainly due to the communication between servers, because applications are compartmentalized in many virtual machines that must communicate with each other. For all these reasons, the network plays an essential role in unifying the services offered in the network infrastructure deployed, achieving flexibility objectives and guaranteed access to services and applications, and ensuring availability, scalability and performance.

Network administrators use management platforms to successfully deliver the offered services. New services generate traffic flows that constantly change between the different points of the network. In order to solve those challenges, network administrators, have used specific solutions, such as improving performance when replacing devices, load balancing, etc.. Those actions imply high costs in the redesign and reconfiguration of the infrastructure. We must take in account that frequently network administrators do not have time enough to design and implement the instantaneous requirements, and therefore, during that time, they do not optimize the infrastructure to offer the services satisfactorily.

For all the reasons explained above, the new networks require a greater degree of automation, integration and architecture design, all of this based on an intelligent network. Software Defined Networks (SDNs) give us the solution to these problems [2]. SDNs present a dynamic, adaptable, controllable and profitable architecture, which can satisfy the requirements of a high bandwidth and the dynamic nature of the services offered by the current networks, allowing having more programmable, automatable and flexible networks as well. The SDN architecture is [3] directly programmable, agile, centrally managed, configured by means of programming and based on open standards.

Adaptive networks are a new class of dynamic networks, whose topologies and states co-evolve. An adaptive network is a network whose links change adapting themselves to their states. That result in a dynamic interaction between the states and the topology of the network [4]. They consist of a collection of agents with processing and learning capacity. The agents are linked together through a connected topology. Those cooperate through local interactions to solve problems of optimized distribution, estimation and deduction in real time. The continuous diffusion of information through the network allows the agents to adapt their performance according to the transmission of data and the conditions of the network; it also results in improved adaptation and performance related to non-cooperating agents [5].

There is a growing interest in the use of video communication systems through Internet and wireless communication networks (such as Wireless Local Area Networks, WLAN, and cellular networks). Unlike conventional data transmission, the transmission of files, especially of compressed video, usually requires high bandwidth. Networks' congestion decreases bandwidth leading to packet loss, propagation of errors as well as delays. This can cause degradation of video quality, freeze frames and generate audio errors, during real-time communication. Video files transfer is often not possible or difficult due to real-time constraints and recovery of losses is not enough to provide required QoE. Video communication systems must also be capable of supporting simultaneous transmissions between two or more places through video and audio. Therefore, there is a need of having an infrastructure and an adaptive network protocol for multimedia communications [6].

The transmission of digital video and audio streams over Internet is part of the audiovisual trends [7]. But it entails the drawback of the need for adequate bandwidth as well as End to End QoS (E2E QoS) [8]. The bandwidth of the customer limits the amount of information of data, and therefore the video that can be received by the user [9]. So it is imperative the use of video compression algorithms to decrease the bitrate of the video stream and get the lowest video size as possible. This measurement is expressed through the compression ratio. The compression is essential in the audiovisual digitization, because large flows of high quality videos cannot be easily handled. There are several studies of compression algorithms for standard television [10], such as the one included in the recommendation BT.601 and high definition television [11]. Moreover, nowadays there is a great variety of devices used to connect to Internet currently [12].

Figure 1.1 shows the results of a survey conducted by Consumer Baromer [13] through Google. It shows the percentage of people claiming to watch online videos daily, through different devices (PCs, tablets, smartphones), and distributed by continents. As it can be seen, the continent with the highest percentage of users is America with 57.1%, the percentages are very similar in Europe and Asia + Oceania, which are respectively 47.5% and 46.8%, while in Africa it only reaches 37.5%.

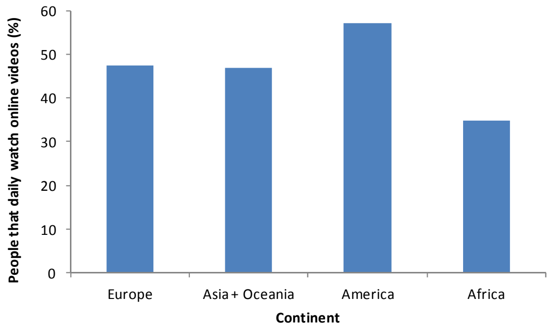


**Figure 1.1.** *Percentage of people claiming to watch online videos daily (data obtained from [13]).*

Figure 1.2 shows the results of the survey in which it can be seen the percentage of people claiming to watch online videos daily via smartphone, and distributed by continents. As it can be seen, the continent with the highest percentage of users is America with a 50%, while the percentages are again similar in Europe and Asia + Oceania. Those are respectively 35.5% and 37.2%, while in Africa it only reaches 25.7%.

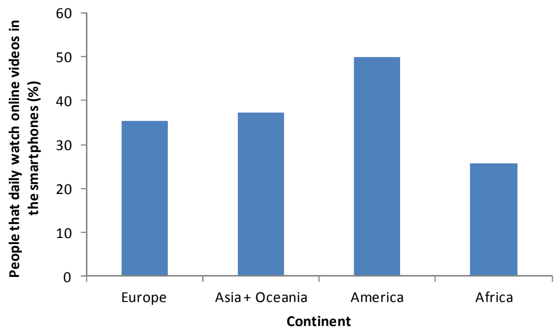


**Figure 1.2.** *Percentage of people claiming to watch online videos daily via Smartphone (data obtained from [13]).*

Figure 1.3 shows four different pie charts. Those show the different percentages of people claiming to watch online videos annually through smartphones. The different colors represent those claiming to watch videos online every day, at least once a week, at least once a month or less than once a month and finally we can see in orange color presents the percentage of users who do not have answered this question.

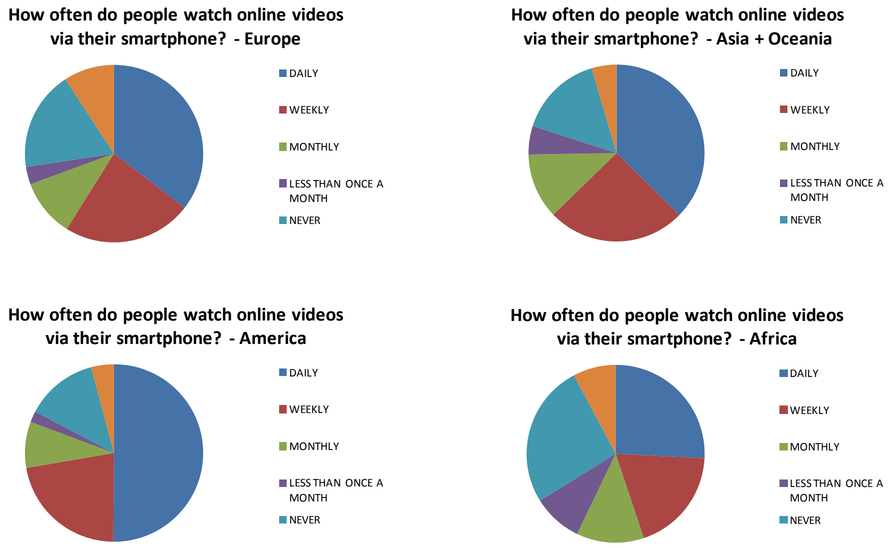


**Figure 1.3.** *Percentage of people claiming to watch online videos via Smartphone (data obtained from [13]).*

The great development experienced by communication technologies in recent years has changed the way we communicate, both in the private sphere and in the professional field. Nowadays, a large part of the world population establishes naturally, communications through different types of devices placed in remote physical locations.

Company's meetings using audio devices only are considered obsolete. Videoconferencing is changing the way we do business and our personal relationships.

You can find multiple videoconference definitions, among them we can highlight:

According to the Real Academia Española (RAE) [14] videoconferencing is the *Remote communication between two or more people, which can be seen and heard through a network.*

In an educational environment, Cabero [15] defines videoconferencing as: *the set of hardware and software that allows the simultaneous connection in real time, by means of image and sound, allows the relation and exchanges information interactively, to people* *who are geographically distant, as if they were in the same meeting place.*

In our scope, when we use the term network we refer to an IP network.

By establishing a face-to-face communication we are able to understand each other more easily and increase empathy between the different interlocutors. This is obvious in environments such as health, where visual contact between doctors and patients improves the results obtained in consultations and treatments.

Videoconferencing avoids possible misunderstandings. Communication is much more efficient and there is a time and costs savings. It must be kept into account that using an appropriated set of tools allows a large number of people participation, which implies the possibility of finding more efficient and reliable solutions in case of facing problems..

The sectors in which the use of videoconferencing can be highlighted are especially:

- Private
- Company
- Freelance professionals
- Public administration
- Healthcare sector - Telemedicine
- Education

It can be basically differentiate two types of video conference based on the used equipment, the video conference with professional devices and that with personal equipment.

In video conference with professional devices, the final equipment is expensive and the rooms from where it is made require special conditions.

In video conference with personal devices, the end devices are usually personal computers or other devices, including mobile devices (tablets, smartphones, etc.). Their connection is made from any location that allows the connection to an IP network.

The correct functioning of a videoconference can be affected by multiple factors. Among these factors we can highlight:

- End-user devices characteristics as CPU, memory, connection interface, display screen size, etc.
- Network interconnection devices characteristics as CPU, memory, queues, interfaces, etc.
- Network characteristics where the transmission is being carried out, such as propagation capacity, bandwidth, delay, variation of the delay, packet loss, etc.
- Characteristics of transmitted information, such as modulation, coding, etc.

There are multiple reasons to use video conferencing, but, in work environments we can highlight that, among other reasons, videoconferencing can be used for the following:

- Save time - It is not necessary to travel to have effective meetings.
- Save costs - It is not necessary to travel to have effective meetings.
- Increase efficiency - We can hold a simultaneous conference with several locations (multi-conference).
- Increase productivity - It gathers all the necessary material to do the work in the place where it has been programmed.
- Increase reliability - Communication depends only on the devices of the users and the network.
- Improve safety - Avoid the possibility of accidents on the travels.
- Increase convenience and cooperation - People can be added to the meeting instantly.
- Face-to-face security - We make sure that our meetings are not publicly known, we do not move.
- Increase employee morale - Traveling to a meeting can be exhausting and makes the willingness of the participants poor.

According to the report Global View: Business Video Conferencing Usage and Trends de Polycom Inc. [16] that was made to more than 1200 decision-makers in the companies, the 96% are convinced that the videoconference was positive in their operations, and more than 76% say they use it at work, even a 56% say they use it at least once a week. In addition, participants who used at that time videoconferencing indicated that the three main advantages as a communication tool were:

- Better collaboration among dispersed colleagues worldwide (54%)
- Greater clarity of the topics discussed (45%)
- More efficient meetings (44%)

Other results presented in the survey indicate that the most used devices for the commercial video conference were:

- Laptops and Desktops (75%)
- Conference Rooms (48%)
- Mobile Devices (42%)

For all the reasons described above, the connection to videoconferencing via mobile devices will be of great importance in the future. This implies the need to previously plan an architecture and protocol, so that the users that participate in it get the highest QoE possible.

### 1.1.1 Applications for videoconference

Currently there are many applications that allow conducting videoconferencing for free. To be able to do it, it is only necessary to have a software application that allows the video call, a terminal with camera and microphone and an Internet connection. Usually the software works independently of the type of terminal, so it can be installed on personal computers, laptops, tablets or smartphones.

Next, we will describe some of the main applications and software clients that support video conferencing.

#### *1.1.1.1 CU-See Mee.*

CU-SeeMe [17] is the first free software that allowed videoconferencing between personal computers in real time. It was installed on Macintosh and Windows PC platforms. Their copyrights belong to Cornell University and their collaborators. The original version was developed in 1992 by Tim Dorcey, with the help of Dick Cogger.

Dorcey published the information about its application in Connexions magazine in 1995 [18].

#### 1.1.1.2 *Facebook (Messenger)*

Facebook (Messenger) also known as Messenger [19] was initially developed as messaging software in the year 2008, being known as Facebook Chat. In 2011 launched an independent iOS and applications for Android. In April 2015, videoconferencing was introduced in 18 selected countries, including the United States, Great Britain, Norway, Belgium, Canada, Denmark, France, Greece, Ireland, Mexico, Oman and Nigeria. It is available for several operating systems: Web-App (Facebook), Android, iOS (Facebook Messenger), and Microsoft Windows.

#### 1.1.1.3 *FaceTime*

FaceTime [20] is a video conferencing application for Apple devices, such as iPhone, iPad, Mac and iPod touch. It was initially announced in June 2010 for iPhone and later it was incorporated into the different Apple devices. It is available for macOS and iOS operating systems.

#### 1.1.1.4 *Google Duo*

The Google application for Google Duo video [21] was presented in May 2016, along with the instant messaging application Google Allo [22]. It is available for Android and iOS operating systems.

#### 1.1.1.5 *ICQ*

ICQ [23] was initially developed in 1996 by the Israeli company Mirabilis as an instant messaging client. In April of 2007 it launches its version 6 in which among other characteristics it incorporates the videocommunication. It happened in 1998 at the hands of AOL [24] and in 2010 to Mail.Ru Group [25]. It is available for many operating systems: Web-App, Windows, MacOS, Android, iOS, Blackberry, and Windows Phone.

#### 1.1.1.6 *Jitsi*

Jitsi [26] was developed in 2003 by Emil Ivov as a project called SIP Communicator at the University of Strasbourg. In March 2011 they added audio and video support using XMPP extensions and it was renamed Jitsi. It is available for several operating systems: Web-App, Windows, MacOS, Linux, Android, and iOS.

#### 1.1.1.7 *Line*

Line [27] was developed by workers of the South Korean company NHN, which is in Japan after the 2011 earthquake, due to the deficiencies that appeared in the telephone services. It is available for Windows, macOS, Linux, Android, and iOS operating systems.

#### 1.1.1.8 *Skype*

The Danish Janus Friis and the Swedish Niklas Zennström designed Skype [28] in 2003, and it was developed by the Estonians Priit Kasesalu, Ahti Heinla and Jaan Tallinn. Both, its code and protocol, are proprietary and closed. Microsoft announced its purchase in 2011, and it replaced Windows Live Messenger in 2013. It is available for the following operating systems: Web-App, Windows, MacOS, Linux, Android, iOS, Blackberry, and Windows Phone.

#### 1.1.1.9 *Tox*

Tox [29] is a peer-to-peer instant messaging and video conferencing protocol. It was developed by the 4chan technology forum community. It was implemented in response to the accusations filed by Edward Snowden concerning Skype, which involved the National Security Agency (NSA) [30]. The first trial versions appeared in February 2014. Its initial developers disassociated themselves from the Tox Foundation in 2015. It is available for Windows, Linux, macOS, Android, iOS, FreeBSD, OpenIndiana, and Sailfish OS operating systems.

#### 1.1.1.10 *Viber*

Viber [31] was initially developed in 2010 in Israel by Talmon Marco and Igor Magazinnik. It was originally presented for iPhone, as Skype competition, later throughout 2012 its adaptation to Android, BlackBerry and Windows Phone was presented. Throughout 2013 its versions for MacOS, Windows and Linux were presented. In February 2014, it was acquired by the Japanese company Rakuten [32]. It is available for several operating systems such as Windows, MacOS, Linux, Android, iOS, Windows Phone, and Blackberry.

#### *1.1.1.11 VSee*

Milton Chen and Erika Chuang, students at Stanford University, founded VSee [33] in 2008. It allows multiple users in different locations to conduct videoconferencing in real time. It is available for the next operating systems: Android, iPhone, iPad, macOS, and Windows.

#### *1.1.1.12 Whatsapp*

WhatsApp Inc. [34] was founded by Jan Koum in 2009. Initially it was a smart schedule for BlackBerry. In 2014, Mark Zuckerberg (creator of Facebook) bought the application to increase users on Facebook. In November 2016, they announced the incorporation of the videoconference service. It is available for operating systems: BlackBerry OS, iOS, Windows Phone, Android, and Symbian.

#### *1.1.1.13 WeChat*

WeChat [35] was initially developed by Xiaolong Zhang in a project of the Tencent Guangzhou Research, which is a research center of the Chinese holding Tencent Holdings Limited. In the year 2011 it was publicly presented in China. Originally it was known in the Chinese market as Wiexin, but when it was presented to the international market in 2012 the name was changed to its current name: WeChat. It is available for Windows, macOS, Android, iOS, and Windows Phone operating systems.

#### *1.1.1.14 Wire*

Wire [36] was presented in December 2014 by the Swiss company Wire Swiss GmbH. Its software is open source (Open Source Initiative). In March 2016, the video call was added among other functions. It is available for Web-App; Windows, macOS, Android, and iOS operating systems.

#### *1.1.1.15 Yahoo! Messenger*

Yahoo! Messenger [37] initially appeared in March 1998 as a client of instant messaging, was known in its origin as Yahoo! Mail. It also served to let its users receive automatic notifications about the arrival of email. Interoperability between Yahoo and Windows Live Messenger was launched July 12, 2006. As of January 2104, it began to support video calls in iOS, later it was extended to other operating systems. It is available for some operating systems such as Web-App; iOS, and Android.

Table 1.1 shows a comparison of some of the most used video conferencing applications. That shows information related to the operating systems used, maximum number of users that can participate in calls and finally the security employed by the application.

**Table 1.1.** *Description of some of the most used video conferencing applications.*

| Client | Operative System | Max. participants | Security |
|---|---|---|---|
| **Adobe Connect** [38] | Microsoft Windows, Mac OS X, Solaris, Linux, iOS, Android, BlackBerry PlayBook | 100 | AES 256, TLS |
| **Avaya Equinox** [39] | Windows, macOS, iOS, Android | Streaming 100000 | AES 256, TLS |
| **BlackBerry BBM Meetings** [40] | BlackBerry O. S., iOS, Android, Windows, macOS | 25 | AES 128 |
| **Blizz** [41] | Windows, macOS, iOS, Android, | 300 | RSA 2048, AES 256 |
| **Blue Jeans Network:** cloud-based videoconferencing service [42] | Windows, macOS and Linux | Standard 25<br>Large Meetings 150 | AES 128 |
| **Camfrog** [43] | Windows, macOS, iOS, Android, | Free version 3,<br>Pro version 8 | HTTPS |
| **Cisco WebEx** [44] | Windows, macOS Linux, iOS, Android. | High Definition (HD) 500<br>High Quality (HQ) 500<br>Standard Quality (SQ) 1000 | AES 256, End-to-end encryption (E2EE), Public Key Infraestructure (PKI) |
| **Cisco Jabber** [45] | Windows, macOS, Android | 100 | AES 256, TLS |
| **Ekiga** [46] | Linux, Unix-like (e.g. BSD or OpenSolaris), Windows | - | |
| **FaceTime** | macOS, iOS | 2 | AES 256 |
| **Fuze** [47] | Windows, macOS, iOS, Android | 25 | 128-bit encryption |
| **Glance Networks** [48] | Windows, macOS, Linux | No | - |

| Client | Operative System | Max. participants | Security |
|---|---|---|---|
| **Google Duo** | Android, iOS | No | End-to-end encryption (E2EE) |
| **Google Hangouts** [49] | Android, iOS, Chrome | 50 | SRTP - AES |
| **GoToMeeting: HD Faces** [50] | WIndows, macOS, Android, iOS | 200 | AES 128, SSL |
| **Highfive** [51] | iOS, Android, macOS, Windows, Chrome OS and Linux | 50 | Web Real-Time Communication (WebRTC) |
| **Jitsi** | Windows, macOS, Linux, iOS, Android | 50 | Datagram Transport Layer Security/ Secure RTP (DTLS/SRTP) |
| **Kopano** [52] | Windows, macOS, Linux, iOS, Android | - | SSL - TLS - Authentication Kerberos |
| **Lifesize** [53] | Browser Support, Windows, macOS, Android, iOS | 50 | SRTP/AES-128, SIP+TLS |
| **MegaMeeting** [54] | Windows, macOS, Linux, iOS, Android | 16 | RTMP / RTMPT / RTMPS, SSL, native encryption |
| **PGi:** iMeet and GlobalMeet [55] | Windows, macOS, iOS, Android | 125 | AES-128, SSL |
| **Polycom:** Pano [56] | Windows, macOS, iOS, Android | 70 | 802.1X, PKI |
| **Signal** [57] | Windows, macOS, Linux, Android, iOS | - | Signal Protocol |
| **Skype** | Web-App, Windows, macOS, Linux, Android, iOS, Blackberry, Windows Phone | 10 | AES-256 |
| **TrueConf** [58] | Windows, macOS, Linux, Android, iOS | 6 simultaneus, up to 250 using multicast | AES-256 |
| **Viber** [59] | Windows, macOS, Linux, | - | TLS, AES |

| Client | Operative System | Max. participants | Security |
|---|---|---|---|
| | Android, iOS | | |
| **Vidyo** [60] | Windows, Android, iOS | - | TLS, SRTP, H.235, AES-256 |
| **VSee** | Windows, macOS, Android, iOS | 15 - 20 | AES-256 |
| **WhatsApp** | Web-App, BlackBerry OS, iOS, Windows Phone, Android, Symbian | 2 | AES-256 |
| **We Chat** | Windows, macOS, Android, iOS, Windows Phone | 9 | AES |
| **Wire** | Web-App, Windows, macOS, iOS, Android, Symbian | 10 | AES 256 |
| **Zoom** [61] | Web-App, Windows, macOS, iOS, Android, Blacberry devices | 50 | TLS, AES 256 |
| **Yahoo! Messenger** | Web-App; iOS, Android | 2 | - |

### 1.1.2 SDN applications for multimedia delivery.

Due to the substantial improvement in terms of hardware and software for computer networks and also the number of network devices available around the world, the interconnection of devices as well as the network complexity has increased. This fact has also enhanced the way we currently process the information transmitted through the network. Over the past 30 years, the Internet Engineering Task Force (IETF) has developed and built around 5500 Request for Comments (RFC) [62]. Nowadays, Internet and other networks are able to offer huge amount of functionalities suited to the requirements of users.

In general, network devices perform two main functions, i.e., the transport function and the control function. On the one hand, the transport function (data plane) that is in charge of sending data through the routes previously calculated. This function is normally performed by specialized circuits known as Application Specific Integrated Circuits (ASICs). The control function (control plane) manages the transport operation thanks to the exchanged information between network devices and the calculation of optimal routes. This allows each device to independently treat the traffic. Network administrators have available few resources to manage and increase the efficiency of the data flows.

A professional of data networks often finds a big challenge the fact of configuring a network and installing the needed network elements to work properly. Due to the services and the features required by the applications currently require, it is possible to increase the network efficiency if we try to manage jointly the entire network. In this way, there appears the need of developing a new technology to reduce the costs and increase the efficiency by automating policy-based flows.

SDN is a new approach for designing, building, and managing networks that separates the network's control and forwarding planes of a better network optimization [63]. The SDN architecture decouples the network control and forwarding functions, allowing the network control to directly become a programmable and the underlying infrastructure to be abstracted from applications and network services [64]. SDN architecture is directly configured programming, agile in its operation, and based on a centralized management. In addition, SDN is open standards based and vendor-neutral. Figure 1.4 shows a SDN architecture.

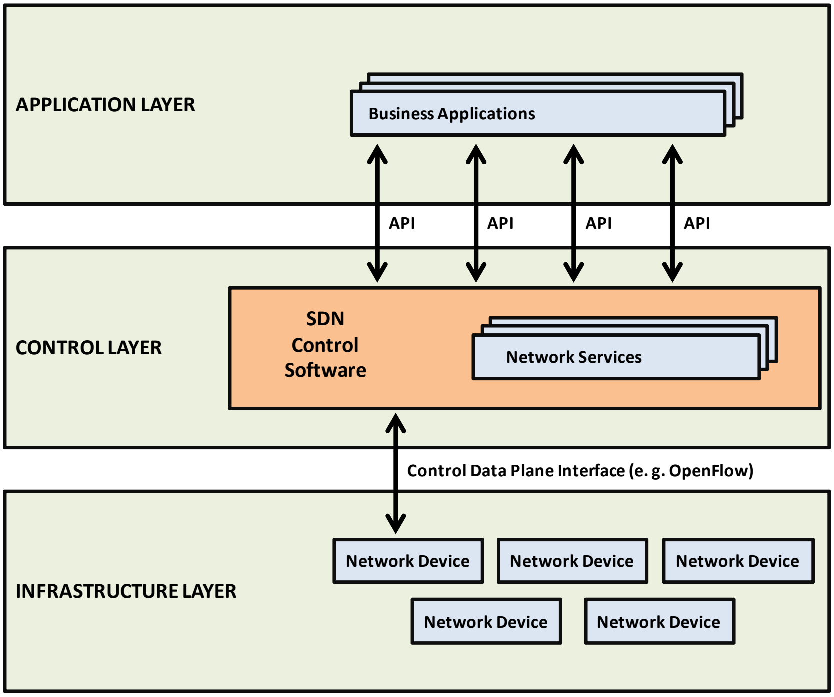


**Figure 1.4**. *SDN Architecture.*

IETF defines the Software-Defined Networking as the set of techniques used to facilitate the design, delivery and operation of network services in a deterministic, dynamic, and scalable manner [65]. We can differentiate two different SDN approaches taking into account their behavior. These are the Open SDN and the Network Function Virtualization (NFV).

Open SDN uses the OpenFlow protocol [66]. OpenFlow is an open standard that enables researchers to run experimental protocols in campus networks. As a result, enterprises and network operators gain unprecedented programmability, automation, and network control, enabling them to build highly scalable, flexible networks that readily adapt to changing business needs [67]. OpenFlow protocol is a foundational element for building SDN solutions.

In NFV, the network devices such as routers, firewalls and load balancers, among others run virtualized on commodity hardware. In November 2012, ETSI created an Industry Specification Group for NFV [68]. Their mission was to facilitate the industry transformation and development of an open, interoperable ecosystem through specification, implementation and deployment experience.

In current networking world, SDN has many advantages. Firstly, SDN centralizes and simplifies the control of the enterprise network management. In addition, SDN offers to network owners and operators more control over their infrastructure, allowing customization and optimization for reducing the overall capital and operational costs. It allows the service providers to generate new revenue opportunities at an accelerated pace through the creation of software based applications for personal computers (PCs), mobile, and Web industries.

As a summary, the most important benefits of SDN is the cost reduction, network overhead reduction, improvements in network management, the decrease of network downtime and the extension of the capabilities of a SDN solution by developing applications to control the behavior of the networking traffic.

SDN has suffered a huge expansion in the last 3 years. Initially, SDN centered its development on the Openflow protocol which emerged firstly in campus environments and data centers. But it was rapidly migrated and adapted to the WAN. Recently, network operators and researchers are including network virtualization in their deployments and research lines. As a consequence of this, nowadays there are many publications about real implementation and proposals [69].

### 1.1.3 Precedents of videoconference

We have reference that the first video conference was made in 1936. It was made using the internal network of the central post office (Postzentralamt Reich) in Berlin and in other cities during the Nazi period. Communication was achieved using a closed circuit of television.

Decades later, the American communications company AT&T (American Telephone and Telegraph) presented at the 1964 New York World Fair (Expo 64) a new service, considered as futuristic, in which calls between users had video and audio. The product was developed by the Bell Company and they called it Mod I Picturephone. Black and white images were transmitted at a frequency of 30 frames per second and communication was established between New York and Disneyland in California. Later they installed equipment in Washington and Chicago that failed in their attempt, due to the excessive cost; but the companies of the sector continued investing in this segment.

Throughout the 70s, of the last century, on the one hand, the service providers of telephone networks began to make facilities to use digital transmission media and, on the other hand, advances in the methods used for the sampling and conversion of analog signals to digital. The problem that arose was that the digital signal required more space for storage and transmission. For example, if the video signal is digitized, using 8-bit Pulse Code Modulation (PCM), with 780 pixels per line, establishing 480 lines per frame, having start from the 525 lines of NTSC (Netware Transmission System Codification) at 30 frames per second, to transmit digital video, transfer rates of 90 Mbps are required. For this reason it was essential to increase the compression of the data.

Using data compression there was an approximate reduction of 50% of the data, obtaining a 2:1 compression ratio (45 Mbps). Previously, 50% of the data was used for synchronization and timing of the television image, and were no longer necessary. By means of T-carrier the transmission of the video could be carried out, using T3 channel at 44,736 Mbps, but it was very expensive. The objective was to use T1 channels, at 1,544 Mbps, in order to have reasonable costs and offer it to the public.

At the beginning of the 1980s, in the last century, the Discrete Cosine Transform (DCT) was used as coding technology, and a compression ratio of 60:1 was obtained. In this coding technology, images are basically analyzed looking for spatial and temporal redundancies.

In the mid-1980s, thanks to the improvement achieved in compression codecs, transmission costs were reduced. In 1984 the company PicTel presented a new coding method called Hierarchical Vectors Quantification (HVQ), which had a compression ratio of 1600:1 (56 Kbps).

In 1988 the International Telegraph and Telephone Consultative Committee (CCITT), formed the telecommunication standardization committee in the International Telecommunication Union (ITU) [70], now known as ITU-T (ITU Telecommunication Standardization Sector) [71]. ITU-T created the Integrated Services Digital Network (ISDN) [72], where the quality standard for videoconferences was regulated, this standard was called H.320.

In the past 90s decade, IP-based videoconferencing was achieved. In 1992, Tim Dorcey, from the Department of Information Technology at Cornell University, developed CU-SeeMe. Originally it was developed for Macintosh and only transmitted video, but from 1994 audio was added to it.

In 1996, the ITU created the H.323 [73] recommendation, named "Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service". This recommendation describes the devices which provide multimedia communications services over packet-based networks (PBN) that may not provide a guaranteed Quality of Service (QoS).

## 1.2 Objectives and motivation

In this work we intend to create a QoE management scheme adaptable to the network for mobile video communications in real time.

The objective of the result of this work is to allow network operators, application developers, service / content providers and end users to develop, implement and use collaborative services, that use bi-directional and multiparticipatory video connectivity in real time in any mobile browser, application, game, device or service platform.

The main objectives of this doctoral thesis are the following:

- Define network adaptive video encoding and decoding algorithms using device-based E2E QoE feedback and, when available, network-based E2E QoE feedback to achieve real-time adaptation according to available device and / or network resources.
- Define device-based and network-based real-time feedback control mechanisms, which can be used to regulate E2E QoE.

- Design, assemble and configure a WAN network for multimedia traffic transmission.
- Implement a test bench to observe both the network traffic QoS parameters variations of the received video and the impact in the image quality due to the network parameters variations.

To carry out the test we have used routers, switches, PCs and an access point. We have obtained measurements from the test bench and we have analyzed the significant differences in order to evaluate the QoS parameters basing it on a variance statistical analysis of the different study cases.

This work attempts to define a mechanism to manage E2E QoE of real-time video, while efficiently utilizing network resources and fluctuating mobile device.

In addition to possible hardware solutions, it is intended to be used by any party interested in implementing applications such as high-quality, real-time bidirectional video chat and multiparty video interaction in any mobile browser, application, game, device and platform. as a complement in all the software.

## 1.3 Precedents

Currently, there are many works about video delivery. Many of them are focused on QoS and QoE. Generally, all the works focus on achieving the best quality of video during at the end user side, in order to obtain the maximum QoE of the final user.

Our research group has been working for a long time on E2E communication using P2P networks, on video streaming and on video streaming in Ad-hoc networks.

Jaime Lloret Mauri defended his Ph. thesis *Arquitectura de interconexión de redes P2P parcialmente centralizadas* (in 27/7/2006). He proposed an architecture to be used in partially centralized P2P file sharing networks, although it can also be used in other types of networks such as for video delivery, where it was applied few years later.

Juan Ramón Díaz Santos defended his Ph. thesis *Design and Implementation of a Communication Protocol to Improve Multimedia QoS and QoE in Wireless Ad Hoc Network* in (22/01/2016). It addressed the problem of multimedia delivery over multi-hop ad hoc wireless networks, and especially over wireless sensor networks.

Alejandro Cánovas Solves defended his Ph. thesis *Diseño y Desarrollo de un Sistema de Gestión Inteligente de QoE para Redes HD y Estereoscópicas IPTV* (in 29/04/2016). He proposed an intelligent management system based on inductive prediction methods to guarantee the QoE of the end-user.

Miran Taha defended his Ph. thesis *Smart Client-Server Protocol and Architecture for Adaptive Multimedia Streaming* (in 27/04/2018). His dissertation is focused on the characterization, design, development and evaluation of different multimedia applications aimed to optimize the QoE.

The research grup has also been involved in many Master Tesis and Degree Projects related with the main topic of this PhD thesis.

In addition, our research group has been part of the Working Group of the Standard IEEE 1907.1, titled A standard for network-adaptive QoE management scheme for real-time mobile video communications. This standard is now in standby, but our group has been actively working on it from 2012 until 2016.

## 1.4 Thesis Structure

After introducing the main issues that have motivated the present Thesis, as well as a description of the main objectives that are pursued, the rest of the Thesis is organized as follows:

In Chapter 2, the state of the art of the improvement of the QoE in the transmission of video and the videocomunicacion is presented. We will present different works related to the improvement of the transmission of video and videocommunication.

In Chapter 3, we present an architecture and several proposals of protocols for the improvement of the QoE of the video received by the end user in different type of networks.

In Chapter 4, we present a protocol and architecture proposal for the improvement of the QoE in videoconference.

In Chapter 5, we include all tests that we have done for the study of the performance and the improvement of the QoE and QoS of the users in the transmission of video and videoconference.

Finally, Chapter 6 draws our conclusion and future research. It also includes the list of publications derived from the PhD.

# Chapter 2. State of the Art of the improvement of the QoE in the transmission of video and the videocomunication

## 2.1 Introduction

Throughout this chapter we will present different works related to the improvement of the video delivery and videocommunication. We will describe different ways that have been used in order to perform tests to measure the quality of the video received. We will also describe how to improve the QoE in Internet Protocol Television (IPTV), and to improve the video delivery in heterogeneous networks using HTML5. Later, we will present a study related to video compression codecs and we will describe different studies on video codecs used in natural environments. Next, we will present the importance of using SDN for multimedia delivery. Finally, a system for the improvement of QoE in videoconferencing will be presented.

## 2.2 Tests to measure the quality of the video received.

In this section we show the most relevant works related to tests to measure the quality of the video received.

In [74], Lu et al. show the potential benefits of a “quality-based” adaptation approach for video applications. They discuss the joint impact of packet loss and encoding rate on video quality, based on which a simple quality feedback-control system has been built. Adaptation is carried out by measuring the quality of the received video and comparing it to a baseline reference. Their preliminary experimental results show both the viability and the benefits of using video quality as the basis of adaptation. Lee et al [75] introduced an IP Multimedia Subsystem (IMS) based testbed which provides a platform for the study of real-time services integration and orchestration. This open-source based testbed is built on the principle of Service Oriented Architecture (SOA), with an emphasis for real-time network services. They developed service-oriented system functionalities such as optical network connection and bandwidth management, mobile client authentication protocols, and video streaming services. They are packaged as interoperable service components, so that they can be integrated and orchestrated through their respective standard interfaces. Finally, they elaborated a proof-of-concept environment via a use-case scenario of having various client-server interactions over a heterogeneous network environment. The work presented by Mu et al. in [76] introduces a system where digital video can be corrupted according to established loss patterns and the effect is measured automatically. The corrupted video is then used as input for user tests. Their results are analyzed and compared with the automatically generated. Within this paper they present the complete testing system that makes use of existing software as well as introducing new modules and extensions. The system can test packet loss in H.264 coded video streams and produce a statistical analysis detailing the results. The user tests can also be used to test the effectiveness of the objective video quality assessment models. Thus, the testbed is designed to provide a common testing platform on which objective and subjective evaluations can be performed with different testing scenarios (such as codec, transmission pattern) with the least human intervention (in terms of set-up and analysis). In [77], Kuwadekar et al present the performance results obtained on an IMS based Next Generation Network test-setup connected to a live operator network. These results are based on the QoE that end user perceive. They can help service operators to deliver content rich applications to the end users as per their expectations. The aim of this work was to understand, with a series of experiment, the behavior of video streaming across various devices and changing network conditions. Through passive measurements, they characterized the behavior of video streaming and evaluated the users QoE.

## 2.3 Systems for the improvement of the QoE in IPTV Transmission.

We have observed that most of the published works focus on redesign and improve the operators' network infrastructure.

Lee et al. [78] present an effective IPTV channel control algorithm that guarantees a seamless channel change and provides a fast data communication service simultaneously via controlling the usage levels of uplink and downlink resources over DOCSIS CATV networks. The proposed algorithm treats the number of adjacent channels and the assigned grant interval as control variables to reduce the network delay part of channel zapping time and improve the quality-of-service of the uplink and downlink. Finally, the experimental results are provided to show the performance of the proposed algorithm.

Zapater et al. [79] proposed an approach that combines QoE and QoS and gives some high level directions ranging from design to management of IPTV service delivery infrastructure. Such directions aim to help service providers to meet customers Quality of Experience requirements and to transition IPTV from small scale deployments to the mass market successfully.

Garcia et al. [80] presented a QoE management system to guaranty enough IPTV QoE to the customer independently of its type of connection (wired or wireless). They compared the delay, jitter, bandwidth and zapping time measurements for IEEE 802.11 b/g and IEEE 802.3u in order to know their differences when QoE parameters are calculated in both networks. The system calculates the user's QoE and notifies him which networks are available for him in order to let him have higher QoE. Then, the user decides to roam to that network. The system allows providing ubiquity to multi-network devices.

In [81], the authors proposed an algorithm that performs the appropriate tasks in the network, based on the measurements gathered from it, to improve the QoE parameter. The proposed algorithm adapts the variations of traffic of an IPTV service, to avoid network congestion and, therefore, packet losses. To show the benefits of our system, they implemented a video transcoder that reduces the bit rate of a video stream. In their tests they have verified that the proposed system achieves a very effective video adaptation, attenuating the burst of video and getting lower bandwidth consumption in the network. Although, this reduction produces lower visual quality, when the reduction in bit rate does not exceed a certain value, it is not perceptible by the user.

Lloret et al. [82] present a network management algorithm and protocol for ubiquitous networks based on the QoE of the end user. First, they explain the developed protocol and algorithm that uses the information provided by the mobile device and the bandwidth, jitter, delay and packet loss of the end user, in order to decide which is the most appropriate video features and the best coding technique to deliver the video streams. A video controller server decides which video server, from a farm of servers, is the best one for each case. They show the algorithm performance when each coding technique is changed in the server due to the transcoding process, and the network performance in terms of jitter, delay and lost packets received by the end user when each coding technique is selected by the video controller server.

Finally, in [83] authors proposed a network algorithm for 3D/2D IPTV distribution using WiMAX and WLAN/Wi-Fi technologies. They propose a network algorithm that manages the IPTV access network and decides in which type of wireless technology the customers should connect with when using multiband devices. These decisions are taken based on the requirements of the IPTV client device, the available networks, and some network parameters (such as the number of loss packets and packet delay), to provide the maximum QoE to the customer. The measurements taken in a real environment from several wireless networks allow us to know the performance of the proposed system when it selects each one of them. The measurements taken from a test bench demonstrate the success of their system.

## 2.4 Video Compression Codecs.

According to Bramberger et al in [84] it is essential to use video compression algorithms that consume little bandwidth and occupy less disk storage space. At present several coding techniques coexist. Some of these techniques are based on the spatial compression of the images (for example, Motion-JPEG), others rely on the temporal compression of video sequences by analyzing the variation of movement between one image and the next (for example, H.261 and H.263). The objective of these techniques is to achieve higher compression ratios while trying to achieve good image quality.

MPEG-4 [85] is an ISO / IEC standard developed by the Moving Picture Experts Group (MPEG). The fields of application of this standard are digital television, interactive graphics applications (synthetic content) and interactive multimedia (World Wide Web and the distribution of video content).

MPEG-4 allows compressing, to a great extent, the audiovisual data for its storage and transmission, while respecting the quality of video and audio. This standard creates encoded representations of the audio and video data that form the sequence through object-based encoding. The standard defines an audiovisual scene as a coded representation of audiovisual objects that have a certain relationship in time and space. These objects are hierarchically organized, and at the base this hierarchy we find objects from primitive communication media, such as: static images (fixed background), video objects (a person speaking), audio objects (the voice associated with a person or background music). MPEG-4 standardizes several of these objects and is able to represent the types of natural and synthetic content. The objects have some attached properties like their spatial coordinates, scale, location, zoom, rotation, etc. These characteristics allow a user to reconstruct the original sequence after decoding all the of object layers. The user can also manipulate the image through simple operations on any of the of the object parameters. It has a facial animation protocol, where a three-dimensional model of a face can be animated in real time. When combined with an audio sample or voice tuner via a text-to-speech converter, the voice can be synchronized with the movement of the lips in real time. The 3D model is not standardized by MPEG-4, only the protocol to control it. It permits to vary the data flow. This can range from 9600 bits/s to 5 Mb/s. The compression is based on the DCT (Discrete Cosine Transformation) with I-frames (keyframe), P-frames (predictive) and B-frames (bidirectional), offering better features than MPEG-1 and MPEG-2 at low data flows.

The MPEG-4 standard is being implemented in video surveillance systems. In this type of systems you need to transmit or store large amounts of video with an acceptable quality, but using few bandwidth or disk space (as the case may be). Depending on the characteristics that you wish to take into account (greater compression, less compression time and computer performance during compression), different variants of MPEG-4 can be used. Codecs behave differently depending on the brightness of the image they are dealing with, that is, the amount of daylight to which the video sequence is compressed.

When the MPEG-4 standard appeared, some software developers programmed codecs that met the specifications of that standard. The most commonly used codecs are DivX [86], Xvid [87] and H.264 [88]. Next we will analyze each one of these:

- **Códec DivX**

DivX appeared in the summer of 1999 (although these same acronyms were previously used for a pay-per-view system). To visualize videos with this codec, its installation is necessary. Among its main features, we can find:

- Four encoding speed modes (performance/quality): Very fast, fast, standard and slow.
- Psychovisual enhancement based on DCT.
- Preprocessing: Spatial and temporal filters.

This codec also allows you to choose the number of I, P and B images and the maximum interval of key images, the global motion compensation the quantization type (H.263 and MPEG-2) and interlaced.

- **Códec XviD**

It originated from DivX 4 and is based on the same libraries from which both DivX 4 and OpenDivX started. It is open source, therefore there are numerous collaborators improving it selflessly. The profiles that XviD has allow the encoding to be compatible with the DivX table players. The "AS" profile (Simple Advance) limits the resolution to 352x288 and the bit rate allows up to 384 Kbits/s. The "ARTS" (Advanced Real Time Streaming) profiles limit the video to the same resolution as the "simple" profile, but the bit rate is 4 Mbits/s. Each profile has several levels with which the resolution/bitrate can be varied.

There is also the possibility of freely configuring the codec without using any profile.

The characteristics that these profiles possess are:

- Number of Images I, P and B. The maximum number of consecutives I, P and B images. Also can select areas where you can apply a different compression setting to a certain group of images.
- Compensation of quarter pixel movement. It helps to assess movement more accurately, refines details and improves compression. It is useful for videos with low resolution. It is not compatible with desktop DVD-DivX.
- Compensation of global movement: It is a technique that reduces the temporal redundancy in a sequence of images.
- Luminance masking or adaptive quantization: XviD reduces quality a bit by using lower bitrates in the brightest areas and in the darkest areas, because the loss of quality in these areas is less visible to the human eye.
- Framed quantization: It allows more compression but it can degrade the image and decrease the speed.
- Quantizer type: H.263 (for low bit rates) and MPEG (for high bit rates).
- Interlaced: Treat each field separately. It is used when the source is interlaced.

The Xvid has the "motion search accuracy" option that other codecs do not have.

- **Códec H.264**

This codec is based on the H.264 standard. It is intended to be used in video applications that have low bit rate (with medium and high definition in television), in video streams over the Internet and in high definition applications in film and DVD. The compression process goes through 7 states:

1. Estimation of movement: In addition to comparing the image analyzed with other previous or later looking for a pattern that indicates the

movement, this codec incorporates the concept of multiple reference of images. It is applied in movements that are repeated periodically in nature, interpretation of movements and obstructions. This reduces the storage size.

2. Intra-Estimation: Used when the object being treated is static. Predict the new block by extrapolating neighboring pixels from adjacent blocks in a group of different predefined directions. The difference between the predefined block and the current block is the encoded information.
3. Transformation: The results of states 1 and 2 are transformed from a spatial domain to a frequency domain with which the codec works. It use DCT based on a 4x4 integer transformation algorithm that reduces the appearance of macroblocks in the video and the "mosquito effect" in the outlines.
4. Quantization: This phase reduces the precision of the set of whole numbers obtained in the previous phase by quantizing them and eliminating the coefficients with frequencie highs without varying the perceptible quality.
5. Filter tour: In this phase a filter called "de-bloking" is used. In the 16x16 macroblocks, the filter eliminates the effects produced by differences in the estimation types of the adjacent blocks or by different quantization scales. In the 4x4 internal blocks the filter eliminates the effects caused by the quantization and by differences with the movement vectors of the adjacent blocks. It is possible to increase the sharpness of the image.
6. Entropic Coding: Entropic coding bases its efficiency on the allocation of very few bits to the symbols that are used very frequently and a greater number of bits to those symbols that appear a few times.
7. Adaptation to the network: Adapt the video data format to the transmission support.

- **Códec H.265**

The H.265 codec is the successor of H.264, it is also called High Efficiency Video Coding (HEVC). It has been developed jointly by ISO / IEC Moving Picture Experts Group and ITU-T Video Coding Experts Group (VCEG).

We highlight the following improvements introduced with respect to H.264:

- Between 40-50% of reduction in the bit rate, compared to H.264 without loss of visual quality.

- Applicable in Ultra HD, 2K, 4K.
- Supports up to 8K.
- Supports up to 300 fps

## 2.5 Systems for the improvement of the QoE for the Transmission of Video in Heterogeneous Networks using HTML5.

In this section, we review some works where a study of different codecs makes the authors decide to use the one that provides the best performance depending on their case.

Lloret et al. presented a study in [89] about the implementation of different codecs for video surveillance to know the best compression codec for this type of service. The work shows the results obtained when the video is compressed with DivX, XviD and H. 264 video codecs. The video was captured at different time of the day for having different luminance. Like this work, our study focuses on a number of codecs but with different purposes. The purpose of the work presented video surveillance while our study is focused on its use in HyperText Markup Language, versión 5 (HTML5).

In [85], Koenen describes the MPEG-4 standard based on the International Organization for Standardization/ International Electrotechnical Commission (ISO/IEC) and shows its features and rules. In addition, it performs a comparison about the possibilities of this compression codec.

In [90], authors demonstrate the importance of choosing a codec when they are working with different environments. They performed multiple types of recordings in different environments where different colors predominate over others because of the characteristic of the place (environment). Measurements were carried out with the codecs MPEG-4, Xvid, DivX, H.264 to study their differences in terms of final size and quality. The difference in our studio with that one is that it does not take into account the devices that can be connected to request the video. The codecs analyzed in that work are different to our work. They were focused on DivX and Xvid, while we focus on the codecs included in HTML5.

E. Ohwovoriole and Y. Andreopoulos compare in [91] two of the major codecs included in HTML5 (MPEG-4, Advanced Video Coding (AVC), H. 264 and WebM VP8) as well as HEVC TMuC. This paper provides the current affairs in state-of-the-art video compression, focusing purely on rate-distortion performance under quantization or bitrate constraints and disregarding system complexity, delay and other domain-specific factors. Moreover, they provide comparative results between MPEG-4 and WebM. In our paper we extend this work to Ogg and we also propose a new algorithm taking in to account them.

Lambert et al. [92] explain the main operation of the MPEG-4 codec H. 264 AVC as a strong option of video coding when the video is going to be delivered through a the network and they compare it with other codecs.

In [93], authors performed a study similar to the one presented by Koenen in [85], but with their own specification of Ogg Theora. Their deployment was different as well as the characteristics of its codification.

In [94], authors explain the future of the video within Internet appreciating the importance of HTML5 for video delivery through its label <video>. It is structured in 3 main sections concerning the background of the video on the Web, the video delivery techniques and its future. It also discusses the possibilities of the API <video> of HTML5 with regard to video delivery but it is not focused on coding like our paper.

Larbier et al. [95] present a work about AVC/H.264 encoding. They used 10 bit color depth, different web codecs and a chroma subsampling of 4:2:2 for modeling broadcast based on this type of compression with loss of information in the peripheral area of the human eye. The results presented were obtained either from ATEME current real-time HD encoders bitwise or accurate and depth software models of upcoming real-time products. Comparisons were made over a very wide range of bitrates in order to illustrate the gains achieved within a great variety of applications. This article is very interesting as it delves into aspects of coding as the color depth or the chroma subsampling. This could be an extension to our study but we focus more on QoS parameters such as the final size (for bandwidth purposes), the encoding time, the frame rate or the final quality. This work only studies a single encoding technique, AVC/H.264, while in our work we included MPEG-4, Ogg and WebM. Moreover, we varied the type of parameters.

This set of new developments can be very useful for different offerings of RTSP video streaming. All of them can be used for video conference or e-learning services because they are real-time services. Although there are many works offering adaptive streaming systems [96] to improve the QoE of the end user, we focus our purpose on developing a system that takes into account the user information to transcode the video to the appropriate codec before it is delivered.

Below, we describe the three main codecs used by HTML5 chosen for the work (mp4, Ogg, WebM). They have been selected because they support either of elements and attributes for building semantic websites, greater compatibility with browsers, platforms or operating systems and also have the coding algorithms more complex and with greater balance between quality, time and size. Now we are going to explain their characteristics as well as their usability for our study.

- **Ogg (Theora)**

It was developed by Xiph.org [97]. The Ogg container format can multiplex a number of independent emissions of video, audio, subtitles and metadata with a video codec (Theora) with losses [98]. Since 2007, it has been incorporated to a large number of software libraries. However, there are some that still do not include it. The codification of the framework of the Ogg algorithm starts with a header called *Ogg Page with* the information and metadata of the media file (version, type of header, position, serial number, and sequence number of page frames or segment mapping). Then it uses segments, which are a group of packets that create a path of data [99].

- **Mp4 (H.264Theora)**

It was developed by the Moving Picture Experts Group and the Video Coding Experts Group. MP4 video codec is also called H.264/MPEG-4 AVC and it is one of the most commonly used codec for recording, encoding and distribution of video [82] [89]. It is a block-oriented motion-compensation-based video compression standard. It typically used for lossy compression. It was created to be capable of providing good video quality at substantially lower bitrates than previous standards.

- **WebM (VP8)**

WebM [100] was developed by Google in 2010 to overcome the problems posed by Nokia, Microsoft and Apple to Ogg Theora. It is based on open source code and placed in a container format derived from Matroska with the video codec VP8, it is currently updated to VP9. Developers are looking for a compression of 50% over the previous version but maintaining the same video quality. VP8 has virtually the same video quality than H. 264 but with a considerable reduction in the space so it is a serious candidate for the codec base.

This codecs review provides a big picture of the major codecs used for the main browsers. Mozilla and Opera have shown his disinterest with Mp4 as the annual fees of H.264 are excessive for them and for the users and, in addition, because it breaks the idea of a Web with open source. Both have implemented WebM and Ogg Theora in their browsers. Apple's Safari and Microsoft's Internet Explorer only work with MP4. Google Chrome is compatible with all three.

There are other codecs such as Dirac, Xvid or ProRes. However, Mp4, Ogg and WebM are elected in shaping Internet standard, because it is used by the major companies working in video encoding. Moreover, they support different compression platforms with FFmpeg or X264 libraries while they have a good balance between coding time, quality, and size.

## 2.6 Video Codecs to use in natural environments

Wild fauna and flora threatened or endangered constitute one of the greatest environmental concerns of the past and present century, whether caused by human or natural origin. The problems that arises from this fact affects the whole ecosystem, both the environment where the species find their existence threatened and the consequences derive such as soil fertility, lack of resources to elaborate medicines or the quality of air and water, among many others [101]. The conservation of the environment is the tool for protecting and maintaining the ecosystem and the monitoring of the natural environment is a small help to facilitate conservation. Climate change refers to the global variation of Earth's climate. This disturbance is mainly attributed to human factors. The causes of this change are very diverse but can be encompassed in three major factors related to each other:

1. the increase of polluting emissions,
2. global warming and
3. greenhouse gases.

The monitoring of the natural environments from which some of these causes derive helps us to obtain relevant data to record these events and to provide solutions to curb or mitigate possible future damage to the entire ecosystem.

The study of species and natural environments and their conservation can be done through the use of sensors or sensor networks which autonomously operate in a controlled way [89]. We can highlight the use of sensors that monitor environmental parameters such as temperature, humidity, radiation, or water levels of a bounded aquatic environment. In the case of animals, their monitoring can be carried out by means of control and surveillance with recording devices, with localization devices or through the use of sensors. In many cases, these sensors are installed directly on the animals, such as tied to the leg of a bird or the neck of a mammal. The recording devices allow direct observation of the behavior of these species allowing both knowing the status of these and acting in case of need. However, sensors do not allow direct observation but they can provide important data to combine with other monitoring mechanisms.

With the current and novel technology, it is possible to introduce monitoring mechanisms in natural environments. The deployment of a sensor network implies in most cases, the use of a wireless technology as EEE 802.11 or Bluetooth to transmit the data and avoid interfering with the environment and wild fauna with cables and other elements [102].

When transmitting data through a network, it is important to know the network limitation. The most important factors to take into account in this aspect are the quality of the connection, speed and available bandwidth in order to ensure a sufficient QoS. The consequence of a low QoS is a poor perception or image quality of the received video. All these factors are directly conditioned to the economic repercussion that the network deployment entails. The economic investment involved in recording devices such as drones or high definition cameras is considerably high, taking into account that these devices are in constant risk of destruction, loss or deterioration.

Because the nature of videos coming from natural environments is enormous and thanks to previous studies [90], we know that processing them in different ways can improve the network performance.

Jiménez et al. present in [90] a study about the most suitable compression codec to recode a video coming from the monitoring of a natural environment. This study takes into account the predominant characteristic color of the video, such as blue in an ocean, the green in a forest or the red in a desert. The rules of the decision algorithm used to select the best codec are based on both the compression time and the quality of the resulting image.

Ferman et al. [103] present a summary of several color descriptors based on histograms to capture and reliably represent the color properties of multiple images or a Gang of Four (GoF)or design pattern. The goal of the study is to provide a solution to the challenge that involves the representation of the chromatic spectrum of the frames or images to make a better administration of visual information.

Lee, design in [104] a parametric transport layer model to monitor the video quality of IPTV services. Authors present the development of a network monitoring tool to evaluate the QoE depending on the physical characteristics of the IPTV system [105]. Based on this study and through this model it is possible to establish a more effective administration, implementation and design of IPTV services.

Zinner et al. [106] propose a method to establish a control mechanism that quantifies the most relevant parameters that influence QoE in video streaming applications based on the H.264 / Scalable Video Coding (SVC) codecs. Through the obtained results, authors design a system that obtains results regarding to the impact of video encoding over the QoE perceived by the user through the resolution of the video or frame rate of the video, using metrics such as Video Quality Management (VQM).

Neogi et al. [107] present a study focused on the compression techniques for interactive video content. Starting from the main goal of offering to the end user additional functionalities while reproducing the video, the authors encounter the problem of storage and transmission costs that are considerably high due to the synchronization of multiple video streams. To solve these limitations, authors analyzed and evaluated different compression techniques.

Video recordings about natural and urban environments have some colors clearly predominant over others. This is mainly due to the characteristic of the environment where the observations are performed. I.e., a video of a forest presents the green color as the predominant one, while in a video of Polar Regions the predominant color would be white, roasted and yellow colors for desert regions and dry farming, blue tones in the oceans, and so on. These video recordings are usually made for documentaries, reports, video surveillance or environmental monitoring, among others.

After these video recordings are performed, they should be encoded to be transmitted through any data network or through Internet to be visualized remotely by any type of device. Phones, computers, tablets are examples of devices that could be used to see these videos. In addition, video files can be stored in physical supports as memory sticks or hard disk [108]. Although the transmission channels are getting better, it is still crucial to encode the videos to improve transmission quality and network performance [109][110]. It is also important to define the best system to encode our videos (as a function of its features) for achieving the maximum optimization of encodings. In our study, the most important factors to be considered when encoding the videos are the scalability and the image quality [111].

The efficiency of each codec depends on the ability of each algorithm to achieve the maximum quality of the compressed image while minimizing the number of bits used to store the data [112]. Based on our measurements, we will see that a codec will be more or less suitable for a particular color when it is able to have higher image quality while using smaller files. So, in order to obtain an optimization video transmission and storage, the election of a codec will depend on the predominant color in the video. IPTV, multimedia and Video IP service providers are aware that any way to decrease the bandwidth wasted per user, will imply saving costs and earn more money. So, any effort to save bandwidth will imply an advance in the industry.

The election of the optimal video codec as a function of the network characteristics and its application is being widely investigated. There are lots of papers that perform studies about concrete codecs or compression systems, but very few of these works take into account the kind of environment that will be recorded and consequently its predominant color. We are going to present some of the most interesting studies on video compression.

E. Dumic et al., present in [113] a comparative study about the efficiency of MPEG-4 Part 10 AVC compression system using progressive and interlaced HDTV (High Definition Television) formats. Authors compared several video quality measures with subjective measure. The tests were performed using three resolution formats. As results show, subjective results of different HDTV formats show that progressive scanning should be considered rather than interlaced for all future HDTV emissions. In addition, authors show that some objective parameters as Peak Signal-to-Noise Ratio (PSNR) and VQM show good performance results in comparison with subjective testing.

Doñate et al. presented in [114] a comparative study of three codecs: DivX, XviD and H.264. They compressed a video captured at different hours of the day to have different video luminance. Authors compared these image sequences based on the compression factor, the time consumed to compress a second of video, % of time using the computer processor and the number of inputs and outputs that the processor have to perform the compression. The results showed that it is necessary to consider different bit rates for each codec. Furthermore, the choice of a codec depended on the limitations or features of the implemented system. Authors conclude that if there are bandwidth constraints in the network, the best codec to be used is XviD, while, if it is required the shortest time to compress the video, the codecs that should be selected are DivX and XviD. Finally, if the limitations are registered in the capture device, the best option is H.264.

P. Lambert et al. present in [92] an overview about the rate-distortion performance of five state-of-the-art video codec technologies in terms of PSNR and JND (Just Noticeable Difference). Authors explain that H.264/AVC's goal of achieving a 50% bit rate savings is more or less satisfied with respect to the DivX 5.1 implementation of the MPEG-4 Visual Advanced Simple Profile when PSNR is used as quality metric. The results of this paper show that in some cases, the use of PSNR and JND as quality metric leads to different conclusions when comparing the coding efficiency of various codecs.

Z. Wang et al. [115] presented an improved version of an algorithm for video quality assessment. Their proposal was based on structural distortion as an estimation of the perceived visual distortion. The algorithm is tested on the Video Quality Experts Group (VQEG) Phase I Full-Reference -TV test data set. These experiments show that VQEG FR-TV Phase I has good correlation with the perceived video quality.

Finally, in [116], H. Schwarz et al. presented the SVC. It is an extension of H.264/AVC. Authors analyze the temporal scalability, the spatial scalability and the SNR scalability of this extension. Through their test, authors show that the H.264/AVC extension for SVC provides various tools for reducing the loss in coding. It provides efficiency relative to single layer coding. They also show the possibility to employ hierarchical prediction structures for providing temporal scalability with several layers while improving the coding efficiency and increasing the effectiveness of quality and spatial scalable coding. Authors also consider the SVC extension of H.264/AVC as a new method for inter-layer prediction of motion. Finally, they conclude that this extension is able to support a modified decoding process that allows a lossless and low-complexity rewriting of a quality scalable bit stream into a bit stream that conforms to a nonscalable H.264/AVC profile.

## 2.7 SDN for multimedia delivery.

This section shows some published works introducing SDN and providing the results of implemented schemes and network topologies.

There are some published works that help understand the reader the concept of SDN and OpenFlow and provide some network implementations. Hu et al. [117] compiled the most important topics about SDN implementation and performed a comparison of different SDN schemes, discussing the future of this research area. Authors focused their work on how SDN separate data and control planes, making scalable networks developments based on SDN. Scott-Hayward et al. [118] highlighted the need of designing security schemes in SDN. Authors classified the security challenges attempting to which SDN layer are affected by. In addition, they proposed some solutions to these challenges and made emphasis on the importance of further work in order to achieve a secure and robust SDN environment, using the capability of being programmed and allowing a centralized network.

Kreutz et al. [119] presented a comprehensive overview of SDN and, despite its fundaments are not new, how SDN could be the new and revolutionary paradigm for Internet. Thus, they did a complete analysis of SDN infrastructure and the main challenges of SDN.

Kaur et al. [120] presented Mininet, an emulator for deploying networks on a single Virtual Machine (VM) as a method to make cheaper tests than tests over physical devices. Authors said that Mininet offered an easy use, performance accuracy and scalability for developing networks. Actually, Mininet has become a powerful tool which makes it easier to test complex networks without the need of having real hardware. In this sense in [121], Paasch et al. evaluated some scenarios for TCP multipath which could derive in huge costs in material like OpenFlow switches.

We can also find some previous works about experiments and performance tests of SDN and Mininet. De Oliveira et al. [122] explained the SDN paradigm, its elements, its purpose and its structure. They also presented Mininet and how it can help researchers to avoid the use of real and expensive hardware for networks. Authors exposed the need of making it easier to implement networks and how Mininet achieves it, with good performance and great reliability.

Moreover, Keti et al. [123] presented an evaluation of Mininet to study its limitations. The results showed that the simulation environment can generate a remarkable effect on the required time for implementing a topology.

In [124], Azizi et al. proposed a new model for measuring the delay in SDN (using Mininet), on a Multiprotocol Label Switching - Transport Profile (MPLS-TP) network. Authors proved that Mininet is not a good emulator for a stress test. However, SDN and Mininet can be used to get other measures as delay values. For instance, Gupta et al. [125] described the design, implementation and the use of fs-sdn (a simulation-based tool) to solve the problem of prototyping and accurately evaluating, at large scale, new SDN-based applications. The results enable the easy translation from virtual environment to real controller platforms like POX and NOX. Authors used Mininet in nearly identical configurations to compare it with fs-sdn.

Panwaree et al. [126] presented an evaluation of the network performance of video streamin over two kinds of OpenFlow-enabled network testbeds. They show the measurements obtained of delay and packet loss when video is streamed over Mininet and Open vSwitch, emulating a network, installed in a PC.

Furthermore, Megyesi et al. [127] introduced a novel mechanism for measuring available bandwidth in SDN networks. They built an application over the Network Operating System (NOS) which was able to track the network topology and the bandwidth utilization over the network links, and thus the mechanism was able to calculate the available bandwidth between any two points in the network. Authors validated their method using Mininet network emulation environment.

We have seen some deployments that use SDN based testbeds for multimedia streaming.

Noghani et al. [128] proposed a SDN-based IP multicast framework in order to control a set of video sources and the impact of SDN on QoE. This framework significantly increased the PNSR of the video, so the user received a good-quality video from a video almost impossible to see. Another practical case is shown by Michael Jarschel et al. [129]. They used a video streaming service like Youtube, in order to evaluate the performance over SDN see how OpenFlow can help us to improve the QoS in video delivery applications.

SDN networks can also be mixed with IP networks. Salsano et al. [130] described the design and implementation of an Open Source Hybrid IP/SDN (OSHI) node. They provided a set of open source tools that allowed facilitating the design of hybrid IP/SDN experimental networks. Their work shows the deployment on Mininet and on distributed SDN, and their test bench. In this way, it is possible to evaluate the costs related with the SDN integration on the Internet.

As we have seen, there are many researches related on Mininet and SDN performance. However, as far as we known, there are very few works showing a performance comparison between real transport protocol over a real network and in a SDN with Mininet. The most similar work is presented in [131], where the Lantz et al. determined that Mininet was not accurate enough by making an experiment. The real hardware network gave different performance than the one emulated by Mininet. This difference was bigger with high network loads due to Linux scheduler, although the authors thought that this limitation is not very important.

## 2.8 Systems for the improvement of QoE in videoconference.

Real-time video connectivity is becoming a key feature in a wide range of services, including communications, corporate collaboration, social networking, gaming and entertainment. The number of these services, as well as the corresponding demand for device and network resources, is increasing at an exponential rate.

The challenges of efficient resource utilization and E2E QoE management are particularly critical in resource-constrained environments, e.g., mobile cellular networks. Due to the increased use of videoconferencing, and anticipating that appear future problems, there is a need to define an E2E QoE Management Scheme for real-time video communication systems.

On January 2012, IEEE announced that, the IEEE Standards Association (IEEE-SA) [132] Standards Board, approved the work group to begin IEEE P1907.1 – Standard for Network-Adaptive Quality of Experience Management Scheme for Real-Time Mobile Video Communications.

IEEE P1907.1 is intended to elevate the quality of mobile video communications and help network operators, application developers, service/content providers and end users to develop, deploy and utilize collaborative services that employ real-time, two-way and multi-party video connectivity within any mobile browser, application, game, device or service platform. The first meeting of the IEEE P1907.1 Working Group was scheduled in March 2012 in Piscataway, N.J.

According to John Ralston, first chair of the IEEE P1907.1 Working Group, "The goal of IEEE P1907.1 is to help raise the bar for real-time mobile video communications, ultimately resulting in a richer and more satisfying end-user experience". "By tackling real-time video capture, transmission and monitoring of received video quality, IEEE P1907.1 is intended to facilitate a high-quality real-time mobile video user experience within the constraints of mobile devices and networks".

Besides possible hardware solutions, this standard further enables any stakeholder to implement applications such as high-quality, real-time 2-way video chat and multi-party video interaction in any mobile browser, application, game, device, and service platform as an all-software plugin. Such an all software solution accelerates both the potential speed and scale of market adoption.

The standard also enables multi-vendor interoperable system solutions that employ real-time network-adaptive video capabilities. It defines an E2E QoE Management Scheme for real-time video communication systems, including those operating in resource varying environments.

In December 7, 2012, IEEE Standards board, selected Jaime Lloret, supervisor of this Thesis, to chair the working group for the Standard 1907.1. He has been chairing the working group till the end.

Figure 2.1 shows the architecture initially proposed by the Working Group for IEEE P1907.1 in their first meeting in February 2013.

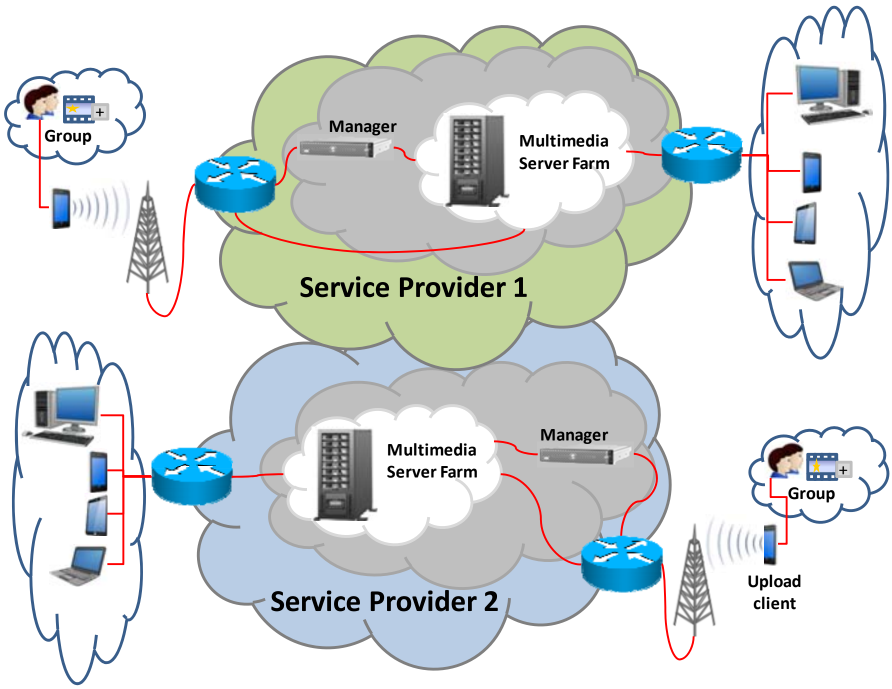


**Figure 2.1.** *Architecture proposed by the Working Group for IEEE P1907.1.*

Stakeholders for this standard include mobile operators, infrastructure providers, and handset manufacturers; PC and tablet manufacturers; other video service providers; video conferencing equipment & service providers; social networks; mobile game developers; browser providers; application developers; cloud computing service providers; business users; consumers; hearing impaired users; and the general public.

At this moment the Working Group for IEEE P1907.1 is in the status of Standby. No group work meetings have been held since 2016.

## 2.9 Conclusion

In this chapter, we have presented the state of the art regarding:

- Relevant works presented in conferences and papers, related to tests to measure the quality of the video received.
- Previous works presented in conferences and papers, focused on designing and improving the operators' network infrastructure..
- Previous works where different authors carry out studies of different codecs, allowing authors to choose the one that provides the best performance according to their application case, focused on HTML5 and in natural environments.
- Previous works introducing SDN and providing the results of implemented schemes and network topologies.
- The implementation of the IEEE P1907.1 standard, to improve the QoE in videoconference.

This review of the state of the art is the basis of the dissertation.

# Chapter 3. Design of an architecture to improve the QoE of the video received by the end user

# 3.1 Introduction

Converged networks use multiservice technologies and include great technical and economic improvements in the delivery of conventional multimedia services [96][7]. Moreover, there have a great variety of devices [12] and many types of traffic flows [133], which QoS should be guaranteed.

Currently, most of the users using IPTV services are motivated by the marketing strategies of the operators. There are service packages, which include IPTV, among others, at a very attractive cost to the subscribers. IPTV is the only platform of totally digital television. Its main advantage is that it allows bidirectional connections. This enables the interactivity and the provision of on-demand services. Their services are different from online or Web TV. IPTV channels are distributed through a service operator, in a closed proprietary network. This allows the operator to control the quality of the video, which would be very difficult if the video has to pass through other infrastructure operators [134]. IPTV consumes a considerable amount of bandwidth, which must be taken into account in both wired and wireless networks [135]. Some authors such as Hossfeld [136] believe that IPTV is regarded as the killer application for Next Generation Networks (NGN).

In the recent years, the global TV market is undergoing a major expansion, due to the implementation of NGN. It is becoming easier to access TV services offered by service operators. This trend is fostered by the increase of users and the number of devices connected to global networks. According to the report Cisco VNI Service Adoption Forecast 2013-2018 [137], in the period 2013-2018, the world population has grown by 1.1 %, while the total number of mobile and Internet users growed 3.5% and 9.2% respectively. Cisco VNI also expected an overall increase of 11% in the number of devices able to connect to the new services. The Cisco VNI also predicts that in the residential sector during the period 2013-2018, the digital video and online TV are the two highest penetration services, which will arrive up to 78% and 86% respectively by the end of 2018. During the same period, the mobile sector has an increase in its mobile video service up to approximately 30%. Finally, in the business sector, the video conferencing service is expected to grow faster.

As operators advance in the implantation of IPTV service, they are observing that QoS and QoE must be guaranteed in order to compete with traditional televisions. Therefore, operators, which are traditionally focused on a QoS management approach, should focus their network management from the point of view of the end user. Here appears the QoE parameter, which is observed from the user's standpoint, on the edge of the network.

In this work, we perform a study in order to gather data and propose a system for improving the end user QoE. The system is formed by an algorithm, which ensures, that after any initiated service connection, the user receiving the transmission will have the maximum degree of satisfaction, according to the type of used device and the available bandwidth. It may be used by operators, to provide a comfortable, stable and economical service in order to achieve the level of QoE demanded by the user.

The main starting hypothesis is that the provision of cognitive mechanisms in combination with SDN improves the performance of current networks that are not adaptive. The routers require complex programming scripts, which causes low adaptability to the traffic that circulates through them, resulting in a QoE for very reduced multimedia services. The main objective is to design and implement adaptability mechanisms following the SDN scheme, to improve the transport of multimedia flows and provide QoE in different parts of the network: the wireless part and the fixed or wired part. For this, we will investigate and develop a new adaptive network that uses forms of prediction of the behavior of users, servers, etc. [138] applying self-learning algorithms, representation of knowledge acquired from the environment and the management information of the network, both wired and wireless. With SDN the network will use its own parameters, to be used in the decision algorithms and improve its performance. In this way we will obtain a network that adapts according to the requirements of multimedia traffic.

This objective is framed in the priority lines of the European calls and lines of action, and how they help the European Patent Office (EPO) to extend its patents worldwide (getting its internationalization). The European Program for the Competitiveness of Businesses and for Small and Medium-sized Enterprises (COSME) 2014-2020 focuses its action on innovation and places Small and medium-sized enterprises (SMEs) at the core of innovation together with entrepreneurship.

Our group is pioneer in this approach [139] (nobody has been proposed previpously proposed an infrastructure or standard until the appearance of IEEE 1907.1). In this way, network traffic can be affected by two main aspects:

User behavior significantly affects network traffic. We must delve our previous studies to observe how mobility [139], the type of wireless technology [83], user preferences [8381] and the type of traffic (IP telephony [140] or Multicast TV [141]), type of device [142], and roaming and vertical handover techniques [143] [144] directly affect network traffic.

The network devices and type of protocols used, due to their traffic patterns and dynamism. Network data requires dynamic routing protocols depending on the state of the network [145]. On the other hand, different video communication applications generate different traffic patterns and use different underlying mechanisms [146], this gives meaning to the use of statistical techniques applied to artificial intelligence to detect and predict the video flows and characteristics of the data from the application layer [147]. Also, many service providers use content delivery networks (CDN) to distribute information within the network according to some predefined parameters [148], since there are patterns of data traffic that can affect network traffic. In addition, the basic technology used in a video communication system is the digital compression of audio and video streams in real time, which greatly affect data traffic [149].

## 3.2 Architecture proposal to improve the QoE of the video received by the end user

Currently operators try to manage their networks by controlling the parameters associated with the video delivery such as delay, jitter and bandwidth. It is essential to control these parameters properly when video is delivered and act rapidly when unexpected things happen. They must react when delay, jitter, packet loss and bandwidth cross a threshold to not affect the QoE.

A fundamental step to ensure a satisfactory QoE at the end user is to analyze properly the information of the network and deliver the video with the most appropriate bit rate. This decision is based on the following assumption: it is better do decrease a little bit the quality of the video than having packet loss, and thus showing errors in the received video or see static images while watching IPTV.

Operators offer different bandwidths for their customers according to the Service Level Agreement. So, IPTV customers may have different bandwidth. We have created several profiles according to the offered bandwidth to the customers and we have established different levels or groups. After classifying the customers based on their access bandwidth, we have defined several groups of customers, which will be assigned to a set of profiles. Then, the video is compressed according to several bitrates which will be assigned to a group of customers in order to provide the maximum QoE satisfaction.

Customers can connect through various types of devices: PCs, tablets, TVs, smartphones, etc. Each device has different screen size, hardware, operating system, browser, etc. Thus, for each type of access, it should be taken into account different types of receiving device profile. Furthermore, depending on the local access (wired or wireless) connection, there are different types of video quality parameters associated. Given these premises, our selection algorithm, should consider the hardware features of the device, its connection type and the video quality characteristics of the service.

Since IPTV channels are delivered via IP multicast groups, customers belonging to a particular profile (of the ones defined above) will be associated to a multicast group. Each multicast group will have assured the necessary bandwidth for video delivery. Thus, the reception of the video will meet the expectations of QoE, and therefore the customer satisfaction. In our proposal we configure as many multicast groups as type of devices per type of network access offered by the operator in order to cover all cases to receive the transmission.

Our proposal focuses on designing, efficiently, and developing a network architecture and its communication protocol, that uses information collected from data patterns and behavior and the traffic patterns (traffic changes, QoS parameters, of data frames, etc.) with the aim of improving the performance of multimedia transmission through the network and adapting according it to each case. The robustness of the network protocol and its applicability to a real environment will facilitate the transfer of our proposal to the Industry. The network architecture as well as the information passing between the different layers is shown in Figure 3.1.

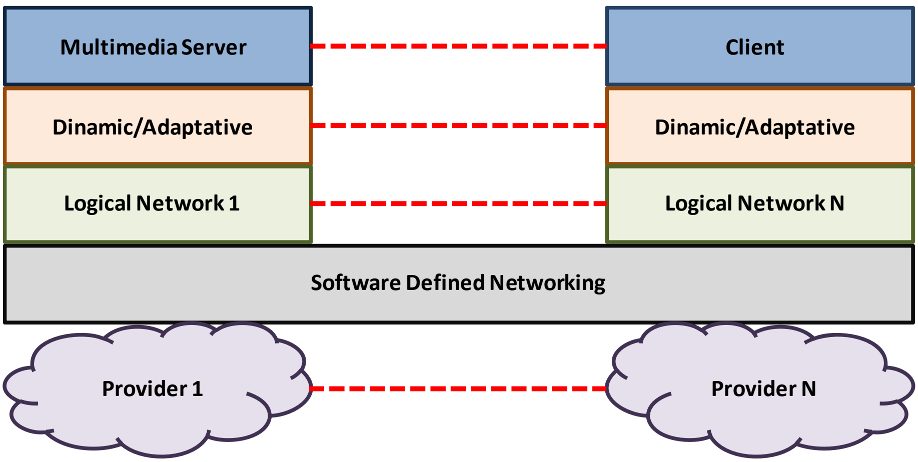


**Figure 3.1.** *Network architecture and information flow between layers.*

To achieve this purpose, a network architecture is proposed where the electronic devices included in it use protocols such as OpenFlow, Floodlight project or OpenDaylight Project, which allow the addition of intelligent algorithms which use the parameters and observed network patterns to warn behaviors empirically. Figure 7 shows how a router can obtain information from other elements of the network, in addition to analyzing the traffic that passes through it. This information is also used by the network algorithm to predict behavior based on this empirical data. The efficient processing of the data and the statistical analysis will allow us to know the traffic patterns [150]. All this network approach will be defined by means of SDN.

The proposed algorithms learn using deductive and inductive techniques on the collected data, and automatically recognize complex patterns [138] in order to make intelligent decisions [151]. The system uses the knowledge taken from the environment, the traffic patterns and the behavior of the users in order to develop a reasoning engine that is aware of what is happening in the environment, what is happening to the users and how operate dynamic network protocols. In addition, he will learn from the consequences of his actions [152]. Therefore, the reasoning engine will be aware of what is happening and will learn from the consequences of their actions in order to improve future decisions. The adaptive network proposal with SDN can be implemented in a wide range of multimedia applications. Figure 3.2 shows the proposed algorithm.

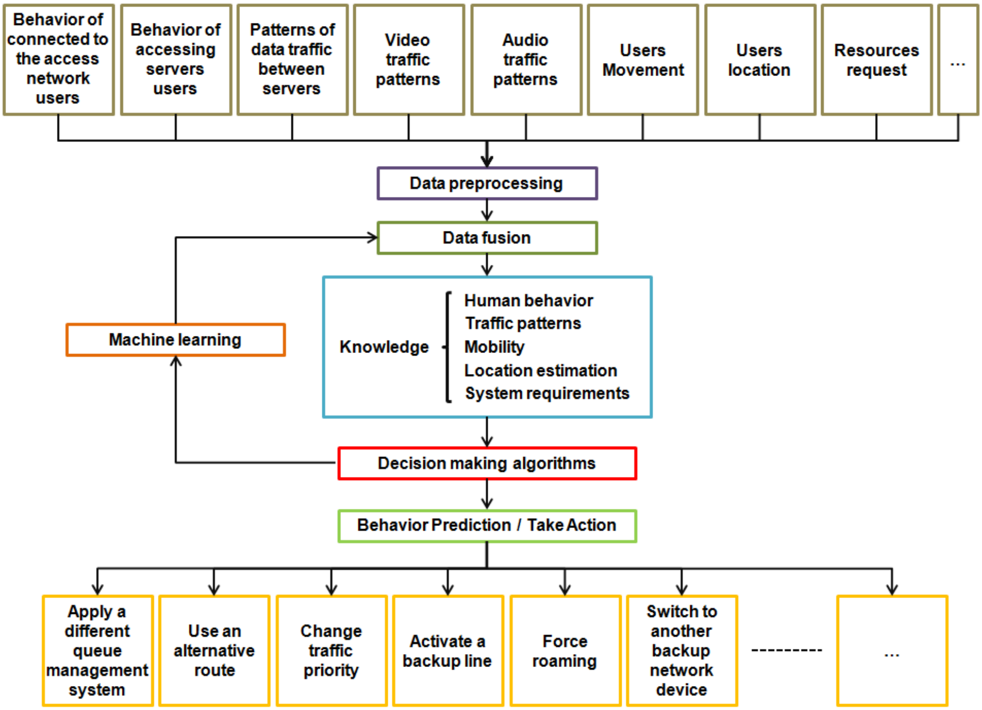


**Figure 3.2.** *Algorithm proposal for adaptive network with SDN.*

In a multimedia network we can distinguish several places where an intelligent system can be added to the network, in order to improve the video distribution and the QoE of the end user. These places can be seen in Figure 3.3.

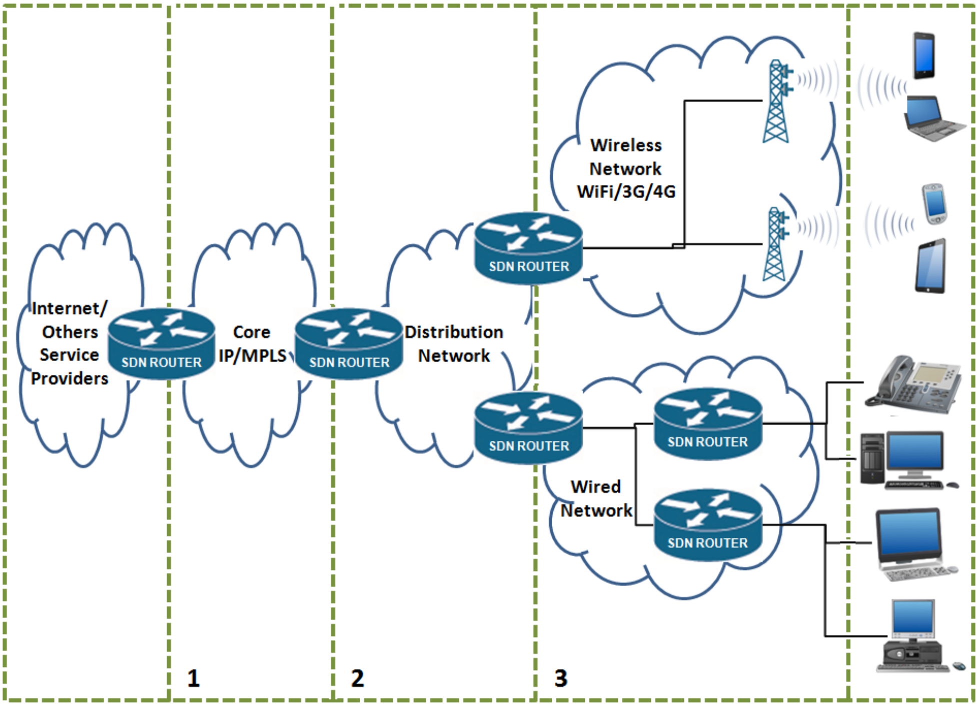


**Figure 3.3.** *Zones where you can put devices with an intelligent system.*

The zones are the following:

- **Zones 1 and 2**: Core network and distribution network. An intelligent system, to improve the quality of multimedia transmission, can be placed between the ISP core network and the customers' access network. The fact of having the intelligent system between two networks allows gathering information of both. Moreover, we can place a server in each service provider in order to collect data from each network. The intelligent system could use the QoS parameters of the network as input parameters. These parameters (jitter, delay, zapping time, etc.) could come from different parts of the network. An example of this type of system was developed by Lloret et al in [81]. In this case, an intelligent system server was placed that compares the parameters measured from the network. If the measured parameters are within a range, the intelligent system does not perform any task, but simply stores the data. If the parameters are not in the range, then the intelligent system must make some changes in the network by sending SNMP messages to the appropriate devices in order to improve video communication. Another intelligent system to improve video communication was developed by Lloret et al in [153]. In this case, the protocol that intermediate devices (routers, switches, servers, etc) need in order to implement a correct video streaming was developed. The intermediate devices carry out various actions according to the control messages and their available resources.

- **Zone 3**: Acccess network. In general, there are several wireless technologies that coexist in the access network. Moreover, all these wireless networks are connected to a common IPTV network infrastructure and they are able to offer an IPTV service. In general, customers have multiband smartphones and several wireless interfaces. Therefore, we can place an intelligent system in the access network to let them know the best wireless network with which to connect. In those situations, where the smartphone can use two or more wireless technologies, the device must be able to measure the RSSI (Radio Signal Strength Indicator) of each available wireless network and send the measured values to the server in order to select the best one. We set out to place an intelligent system in the access network. Each time a device joins a wireless network, it sends the MAC address and SSID, from the detected access points, and the QoS parameters to the server that has the intelligent system. The server, in which we have the intelligent system, has a database with all the wireless networks of the access network, and intelligently chooses the best wireless network for each smart phone or IPTV receiver [83] [140][141]. It is a dynamic system, in which the decisions are based on the values of the parameters. This system is intended to provide the highest QoE of the end user.

With the proposed system, multimedia communication networks will be more efficient. An example of the proposed system is shown in Figure 3.4. The fact that devices can collaborate within the same network of a service provider and even between devices from different providers can improve the quality of end-to-end multimedia communications. The different networks based on groups and the protocols proposed in [148] [154] demonstrate the benefits of existing systems.

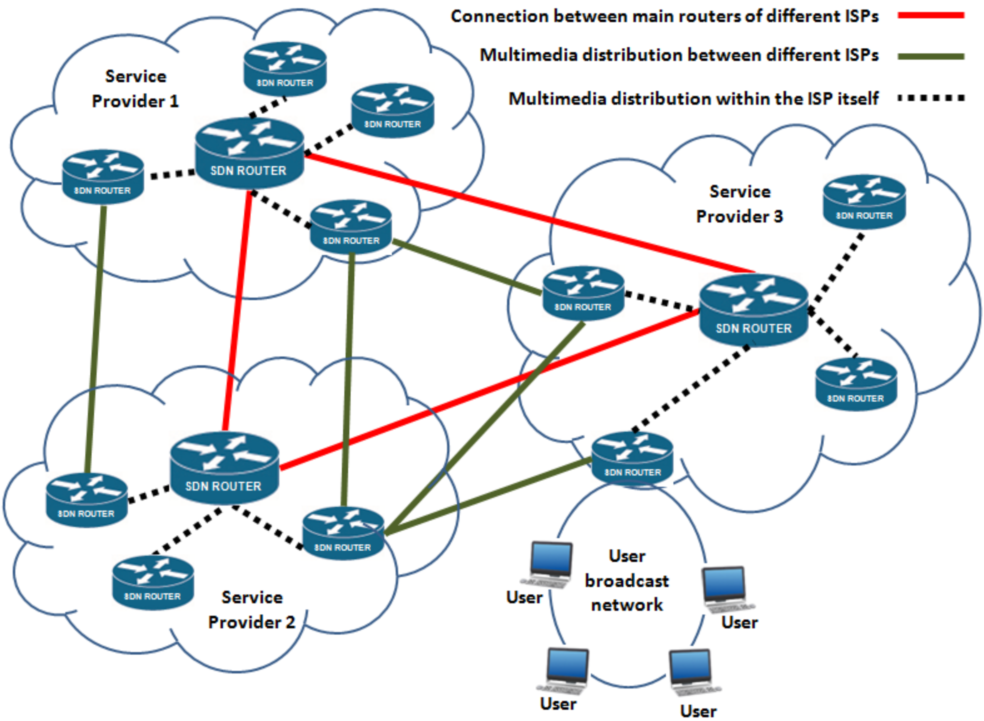


**Figure 3.4.** *Example, network with several collaborating service providers.*

### 3.2.1 Protocol for IPTV

The communication protocol designed for the proper assignment for a customer is shown in Figure 3.5. There are three parties involved in the exchange of messages: the customer connecting to the IPTV, the server providing the authentication and selecting the multicast channel, and the IPTV server. Both servers are located at the IPTV service provider side.

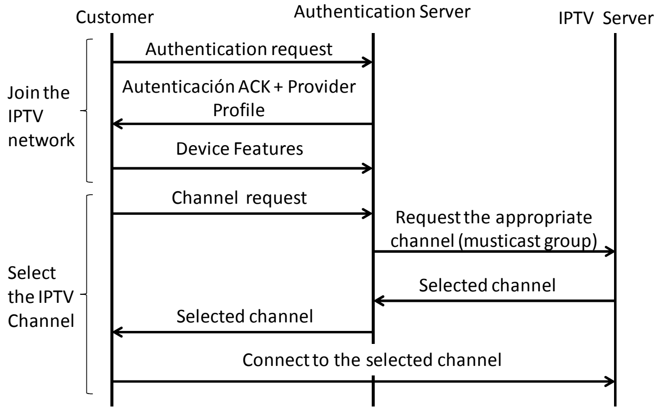


**Figure 3.5.** *Communication protocol for channel selection.*

When a customer wants to connect to an IPTV service, there must be an authentication process. So, the customer's will send an "Authentication Request" message to the authentication and selection server. If the client has the proper contract with the IPTV service provider, it will reply with the "Authentication ACK + Provider Profile" message to the customer.

We define Client Optimization Parameter (COP) as a data structure that will allow the new architecture to classify the flows of the data that will be transmitted. The nodes that take part in the video transmission use COP to agree a level, so that the transfer is made using the minimum amount of resources necessary while obtaining the maximum expected QoE in the users.

The number of COPs defined must have the necessary elements to cover the most common multimedia flows, which are essential to establish a correct visualization between users. Although to make an agile management, the amount of COPs should not be very broad.

Each COP has a hexadecimal one-byte long code, and an alphanumeric code called ACode whose size is variable.

COP has the following parameters inside: Operative System, CPU, RAM, Screen Resolution, Screen Orientation, Connection Type, available Bandwidth, etc.

Then, the customer will send a message with the characteristics of the device (COP) performing the connection to the authentication server, which will allow the server providing the authentication to request the most appropriate IPTV multicast channel based on these characteristics. The message sent to the IPTV server is called "Request the appropriate multicast channel".

When the server selects the channel for the customer, it sends a message called "Selected multicast channel" to the authentication and selection server, who will be responsible of forwarding it to the customer. When the customer receives the channel to be associated, it will send the message "Connect to the selected multicast group" and, when it is accepted by the server, it will be associated to the appropriate multicast group according to its device and contract with the service provider. From that moment, until the customer disconnects, it will receive the transmission through a multicast channel according to the selected profiles. If the customer initiates a second device with different profile in its home network to the same musticast channel, but requiring higher bandwidth, both devices will be associated to the multicast channel with higher bandwidth requirements, and thus the higher profile features, which will allow to serve both devices with a single multicast channel.

If the customer does not receive the assigned channel in a short period of time, it will attempt to obtain it again after 5 seconds. In case of exceeding the maximum number of attempts trying to receive the channel assignment, it will pass to the disconnection state. The customer can restart the process again if he/she still desires to connect to the IPTV service.

The protocol will avoid any network congestion because each customer has enough initial bandwidth. In all cases we consider that the network is operated by a provider that ensures QoS when serving IPTV, and then, the customer will associate to a group that its features will be sized according to the contract with the IPTV service provider and the operator.

Figure 3.6 shows the algorithm designed, which is based on the protocol described above. By using this algorithm, an operator would ensure that it will offer the customer to deliver IPTV channels at the highest quality according to the device characteristics used to watch IPTV, while it is also adjusted to the bandwidth hired by the customer. Therefore, the user will receive the highest QoE according to its device and network access features.

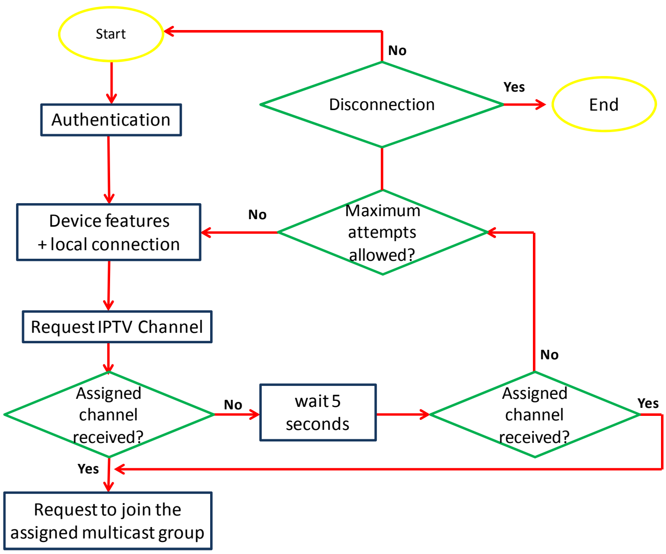


**Figure 3.6.** *Proposed algorithm to improve IPTV QoE.*

### 3.2.2 **Protocol for Heterogeneous Networks using HTML5**

There are many people involved in the development of HTML specifications and implementations in web browsers. Since the first intention to include the <video> element in HTML5 in the year 2005 up to the unanimity of all the major browsers, with the latest addition to Internet Explorer, it took 9 years. The <video> tag in HTML5 became the first native implementation of multimedia support in the browsers without the need for third-party plugins such as Quick Time, Microsoft Windows Media or Flash [155]. There has been a debate for a long time in which it was discovered the need for so-called codec base, a standard encoding format for HTML5, which should be compatible for all browsers. Although any codec can be encapsulated in a container, it can only be included in specific containers with customized mappings [99]. This is the main connection between the multimedia compression and the HTML5 standard, which later facilitated the emission of videos and the support necessary for the codecs in the rendering engine of the browser through their API (video, Canvas, SVG).

The advantage of HTML5 is that the browser must include in its rendering engine the audio/video player engine. Therefore, if the browser supports to codec it is because it is implemented through a bookstore and an external plugin is not needed. Before HTML5, users had to install codec packs to be able to play a video in both Internet and local players. HTML5 simplifies the code to perform video streaming by using a semantic language, expanding the possibilities with their APIs and using an easy integration with JavaScript. The main differences with third-party plugins include: less loading time of the web pages, increases compatibility with more browsers, includes more options (more dynamic pages with less code) and allows the use of the website offline.

We present a study and analysis of the characteristics of the video compression codecs included in HTML5. We included in this study several parameters such as the type of browser, frame rate, bitrate, encoding time and final quality of the video. It let us design an algorithm which is based on the received parameters, decides which is the best codec for that case and which transcoding should be performed without the need of a buffer at the end user side since its codification time is lower than the playback time [82].

Let's describe the proposed algorithm, which uses real-time transcoding in order to stream video in HTML5.

The whole process starts when a user requests to see a video using any type of device and using any type of browser. The device sends the request to the server using http protocol. If the request is successful, the server will take into account the information of the user and device (browser, Internet access, etc.) and, then, the server will deliver the most appropriate video (transcoding it if it is necessary). However, if the request is not successful, it will disconnect the user and the stream of video will not be sent. If the server accepts the connection, via the standard label HTML5, it analyzes the used browser through a script as well as the geographical area of the IP address. The IP address will help the system to know the destination country and, therefore, the frame rate used in that country. A JavaScript script will be in charge of estimating if 29.976 fps should be sent for NTSC systems, or 25 fps for the rest of the world (PAL or SECAM). Our previous studies show that the coding should be done from an original file with a bitrate of 1536 Kbps up to 7950 Kbps in order to maintain high video quality and the time required for transcoding.

In figures 3.7 and 3.8 show the operation of the overall process from the request of the user till the video is delivered from the server, depending on the parameters and the characteristics of the user. Figure 3.7 shows the operation process from the user perspective. It sends an authentication request to the server and waits until it is confirmed. If accepted, the device sends a request to the video server using its web browser, through the HTTP protocol, with its profile. The device profile stores information such as its IP in order to find the geographical location and thus estimate whether PAL, SECAM or NTSC should be used. Then, the customer waits for the reply from the video server. The number of requests is limited to 5 times. Then, the system restarts. If the connection is accepted, it waits for the video streams which are stored in buffer during 300ms and starts playing it.

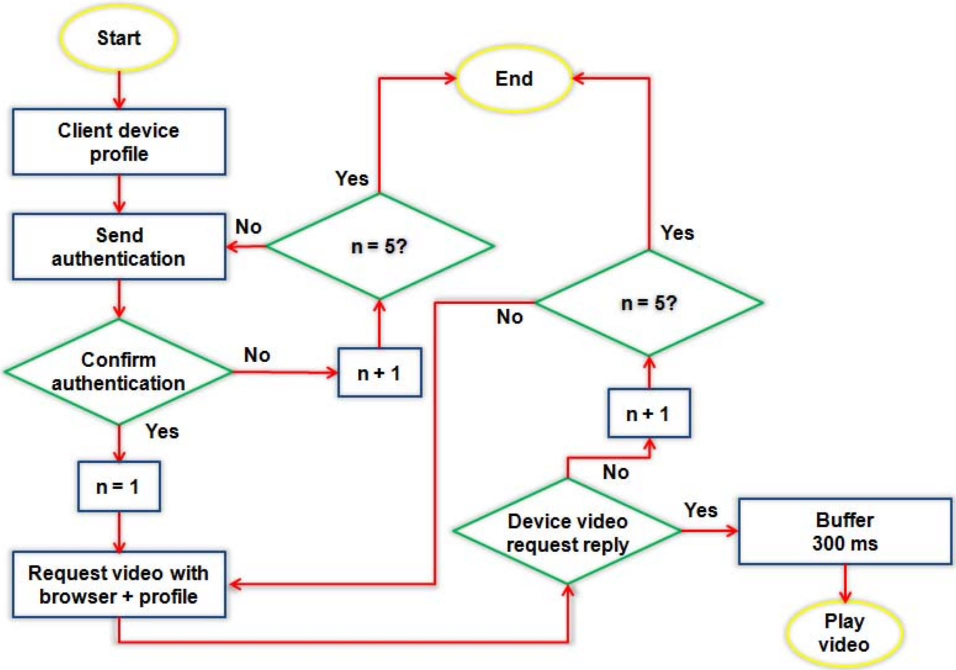


**Figure 3.7.** *Algorithm performed by user requesting video using HTML5.*

Figure 3.8 shows the algorithm used by the video server. When the server receives a connection request, first it authenticates the user. Then, it confirms that the user is requesting video using HTTP and reads its profile. The system uses a counter which is used to stop the connection request and release all resources when the number of continuous connections from the same device exceeds a threshold. If the video request is accepted, it is carried out the detection of the type of browser from which the request is performed and the profile information of the client device is saved. Then, it transcodes the video according to the used web browser, the appropriate fps (estimated by using the user IP) and the appropriate bitrate for the user's network access. Finally, the server delivers the appropriate video for that user.

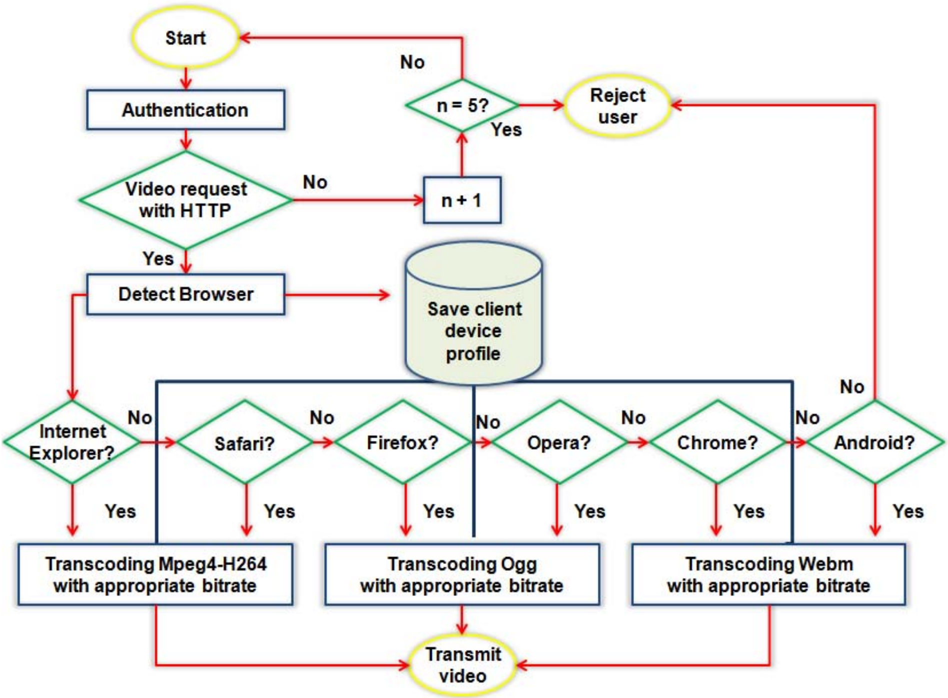


**Figure 3.8.** *Algorithm used at the server side to deliver video in HTML5.*

A multi-user scenario of a service using multicast video delivery and a configuration using a single-user scenario is shown in Figure 3.9.

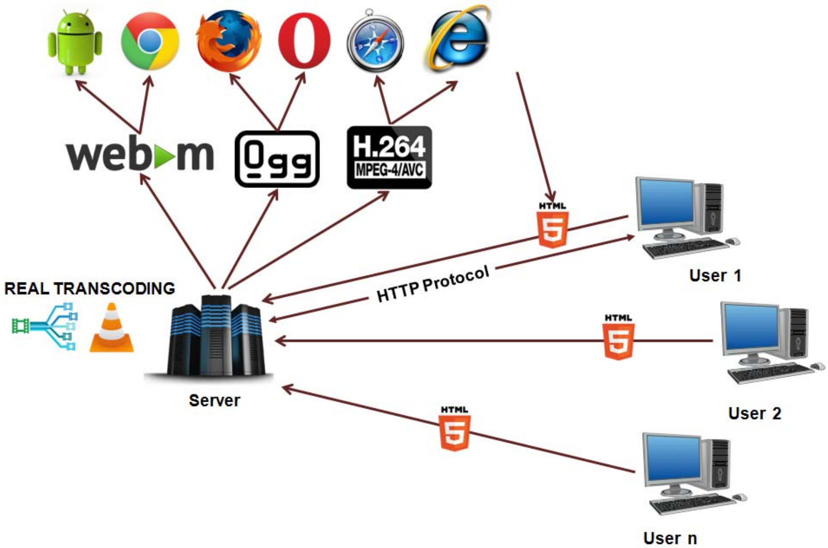


**Figure 3.9.** *Overview of the system procedure when a user requests video.*

To achieve this, various customer devices would agree to view the required video and, after an authentication process, the server will transcode the video according to each type of device and delivers them through different multicast groups. When a new device joins the system and the appropriate video stream is yet being delivered for other device (even from other customer) it will join that multicast group, thus saving bandwidth for the operator. This would imply a high processing capacity in the servers. An example would be an e-learning platform where different students connect to a server where there is a professor is performing a videoconference. This system will allow delivering video to each student in a personalized way, depending on his/her browser, operating system, available bandwidth or framerate (PAL or NTSC).

### 3.2.3 Selection of the best compression codec depending on the environment

We have done a comparative study about the dependency of the video compression codecs with the type of video content (The study is presented later in section 5.7). In this case, we will focus our study in the chroma video characteristics. It includes videos with red, green, blue, white and black as a predominant color. From this study, we propose an intelligent algorithm and protocol to decide, considering the predominant color, which codec (MPEG-4, DivX, Xvid and H.264) will be the optimal one for each video.

In this section a new algorithm is proposed. Video codec for multimedia transmission is dynamically selected by analyzing the RGB histogram and the QoS and QoE constraints. Then, the network protocol in charge of the synchronization between the sender and the destination is explained.

#### *3.2.3.1 Algorithm proposal*

In order to improve network performance and the user quality of experience for video delivery we have designed an algorithm. This algorithm takes account the network parameters such as QoS constraints and video properties like the prevailing color. Fig. 3.10 shows the flow diagram for the proposed algorithm.

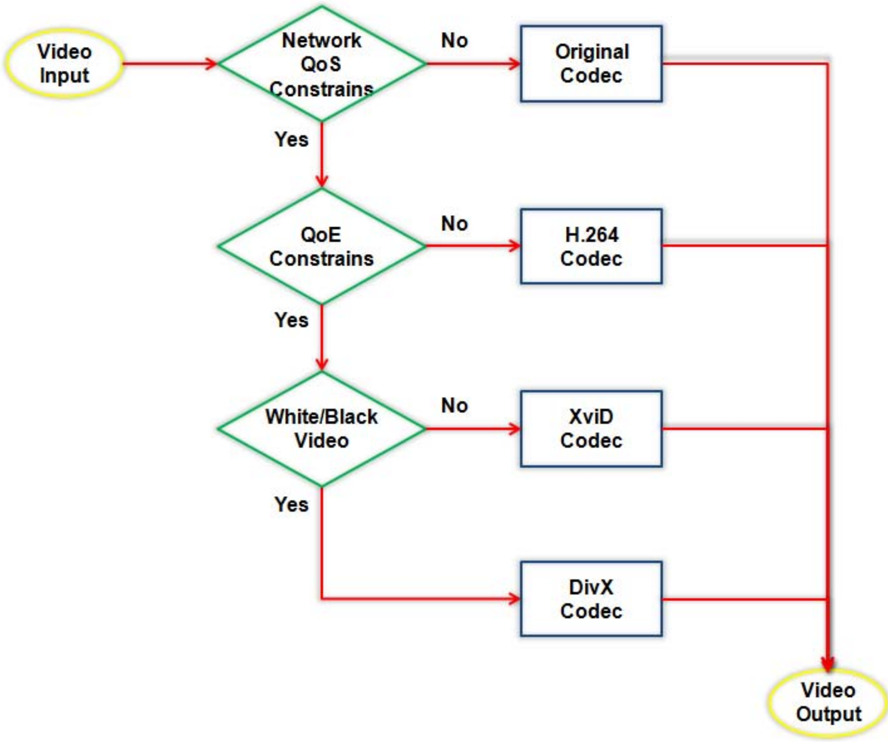


**Figure 3.10.** *Decision Algorithm for Codec Selection.*

Video transcoding is an intensive consumption time process. It only can be justified when the latency introduced is offset by the QoS or QoE improvement. Thus, in the first step of the algorithm, the current network performance is checked; the system looks for QoS constraints such as available bandwidth, RTT values and lost packets counters. Trigger thresholds have been established for each relevant network QoS parameter. If there are not network QoS constraints, the video streaming is just transmitted using the original input codec, so no transcoding is made. If any QoS parameter reaches its threshold value, then the QoE requirements are analyzed. If it is not required a high level of Video Quality, then H.264 part 10 is selected. This codec provides the best compression rate but the worst video quality. For any other kind of video communication where QoS and QoE must be considered, the color spectrum needs to be analyzed. XviD and DivX codec can offer the best ratio compression but we have found that they are a bit different as a function of the chroma characteristics of the video. The achieved video quality of the video signal, measured by QoE standards, is also different for each combination of codec and chroma signature. Taking into account the preliminary experimental results, the next stage of the algorithm evaluates if the video is black and white or if the prevailing color is white or black. If it is true, DivX codec is selected. Otherwise, when red, green or blue are the prevailing colors or it is a color film, XviD is selected.

#### *3.2.3.2 Protocol proposal*

Network QoS constraints or video chroma can change over time. They are periodically checked at the sender side. Then the algorithm is executed to decide if the codec has to be changed. We need to design a network protocol to manage the synchronization between both sides of the video communication. This protocol has to be efficient and provide low overhead. Moreover, the protocol must cover all cases. Designed protocol messages are encapsulated on the UDP protocol and ACK messages have been established for each control message.

Table 3.1 shows the list of protocol messages that were defined. The HOST-SYNC message is sent by the node receiving the video before starting the video communication. This message carries the required information to manage the codec selection: Supported codec, QoS thresholds, QoE expectations and the original codec. The video server answers sending a HOST-SYNC-ACK message which includes similar information from the video server side. When the decision algorithm establishes that a new codec has to be used, the video server uses the CODEC-UPDATE message to synchronize with the node playing the video. This message informs about the new codec and the starting point for the new transcoding.

**Table 3.1.** *Protocol messages.*

| **Software** | **Video Compression** | |
|---|---|---|
| | **Host** | **Description** |
| **HOST-SYNC** | NODE | ıchronization message from the node to the server |
| **HOST-SYNC-ACK** | SERVER | K message of the HOST-SYNC message |
| **CODEC-UPDATE** | SERVER | date information about the codec selected by the decision |

| Software | Video Compression | |
|---|---|---|
| | Host | Description |
| | | algorithm |
| **CODEC-UPDATE-ACK** | NODE | K message of the CODEC-UPDATE message |
| **CODEC-RESET** | SERVER | ;et codec information and come back to the original codec |
| **CODEC-RESET-ACK** | NODE | K message of the CODEC-RESET message |
| **HOST-END** | SERVER | ›se the communication |
| **HOST-END-ACK** | NODE | K message of the HOST-END message |

In order to avoid a failure in the synchronization process, the change to the new codec only takes place when the video server receives the confirmation message, called CODEC-UPDATE-ACK. The CODEC-RESET and CODEC-RESET-ACK message allow restoring at the original configuration. Finally, the HOST-END message is used to stop the communication when the video streaming is finished. The node playing the video answers with the HOST-END-ACK message and the control communication ends. Figure 3.11 shows the message flow diagram.

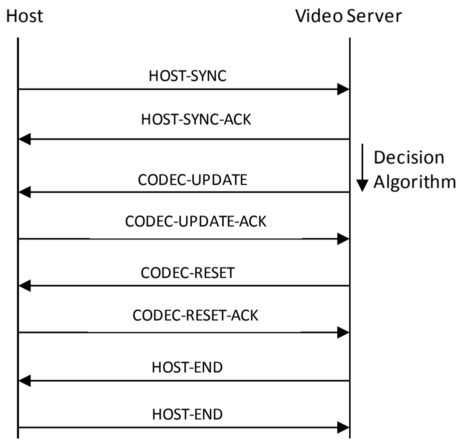


**Figure 3.11.** *Message flow diagram.*

### 3.2.4 Selection of the best autonomous video compression system for environmental monitoring

We are going to present an autonomous video compression system for environmental monitoring. The system is based on a server that executes our algorithm developed in Python and MATLAB® [156] and analyzing the network constraints and the predominant color of videos provided by cameras. The algorithm analyzes the RED-GREEN-BLUE (RGB) components of the video and performs the transcoding tasks.

We will deal with a description of the different studies carried out for the development of the work. Figure 3.12 shows the process carried out to carry out the preliminary study for the characterization of each of the resulting videos and the computation time to recode the videos.

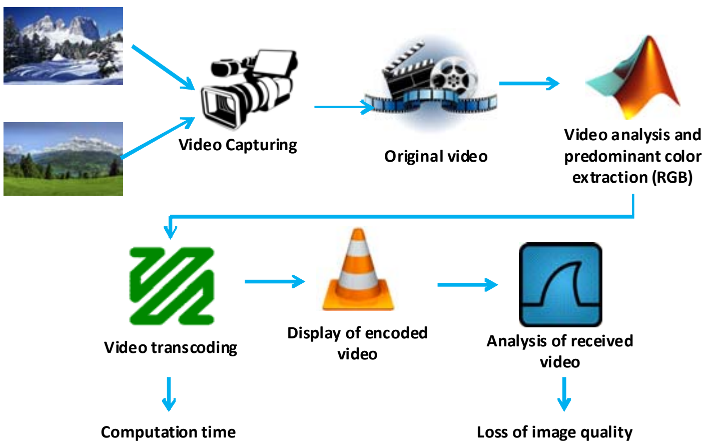


**Figure 3.12.** *Description of video transcoding process.*

#### *3.2.4.1 Video analysis and extraction of predominant color*

In order to determine which codec best fits a video recorded in a natural environment according to its average RGB level and its predominant color, a short algorithm has been developed that quickly and efficiently determines which ones are the color parameters of a video desired for this study. We proceed to the description of it.

To determine the predominant color, five steps or tasks that detail the algorithm in a structured way have been established:

1. Read a video stored in the system using the VideoReader function. This function creates a processable object associated with a file that contains a video. Depending on the platform where MATLAB® is running, some formats or others will be valid. Being the platform where the present Macintosh work has been developed, the supported video file formats are: MPEG-1 (.mpg), MPEG-4 including video encoded in H.264 (.mp4, .m4v), Apple Quick Time Movie (.mov), 3GPP, 3GPP2, AVCHD and DV.

2. The necessary analysis parameters are obtained to later process the video. The Height and Width functions return the resolution of the video, that is, the height and width of the set of frames measured in pixels. Next, the number of frames of the video is obtained by applying the Ceil function, which returns the approximate integer closest to the obtained value, the product of the duration of the video (obtained with the Duration function) and the frame rate per second (obtained with the FrameRate function).

3. An auxiliary matrix 'and' filled with zeros with the size "Number of frames x 3" is created, since for each frame we will have a red 'R' value, a green 'G' value and a blue 'B' value predominant.

4. The main loop of the algorithm has the function of traversing the video file frame by frame. For each iteration of said loop, an internal loop is executed that runs through each row of pixels of the frame, storing in three auxiliary variables the summation of each color. At the end of the internal loop, the mean of the RGB level of said frame is calculated and stored in 'y' and passed on to the next one. Thus, the result of this loop is the obtaining of the complete matrix 'y', which will contain all the average RGB levels of each frame.

5. Finally, the RGB average is calculated from the 'y' matrix, obtaining the variable RGB_resultado, which will contain the average RGB level of the whole video object, being able then to observe the predominant color.

Figure 3.13 represents the operation diagram of the automatic calculation algorithm of the RGB values that is essential to perform a previous classification of the videos.

In order to test the correct functioning of our algorithm, different executions are carried out where different colors predominate. It is important to note that, for the RGB level, pure colors are considered:

• Pure red chroma: RGB level = R (255) -G (0) -B (0).

• Pure green chroma: RGB level = R (0) -G (255) -B (0).

• Pure blue chroma: RGB level = R (0) -G (0) -B (255).

• Pure white chroma: RGB level = R (255) -G (255) -B (255).

• Pure black chroma: RGB level = R (0) -B (0) -G (0).

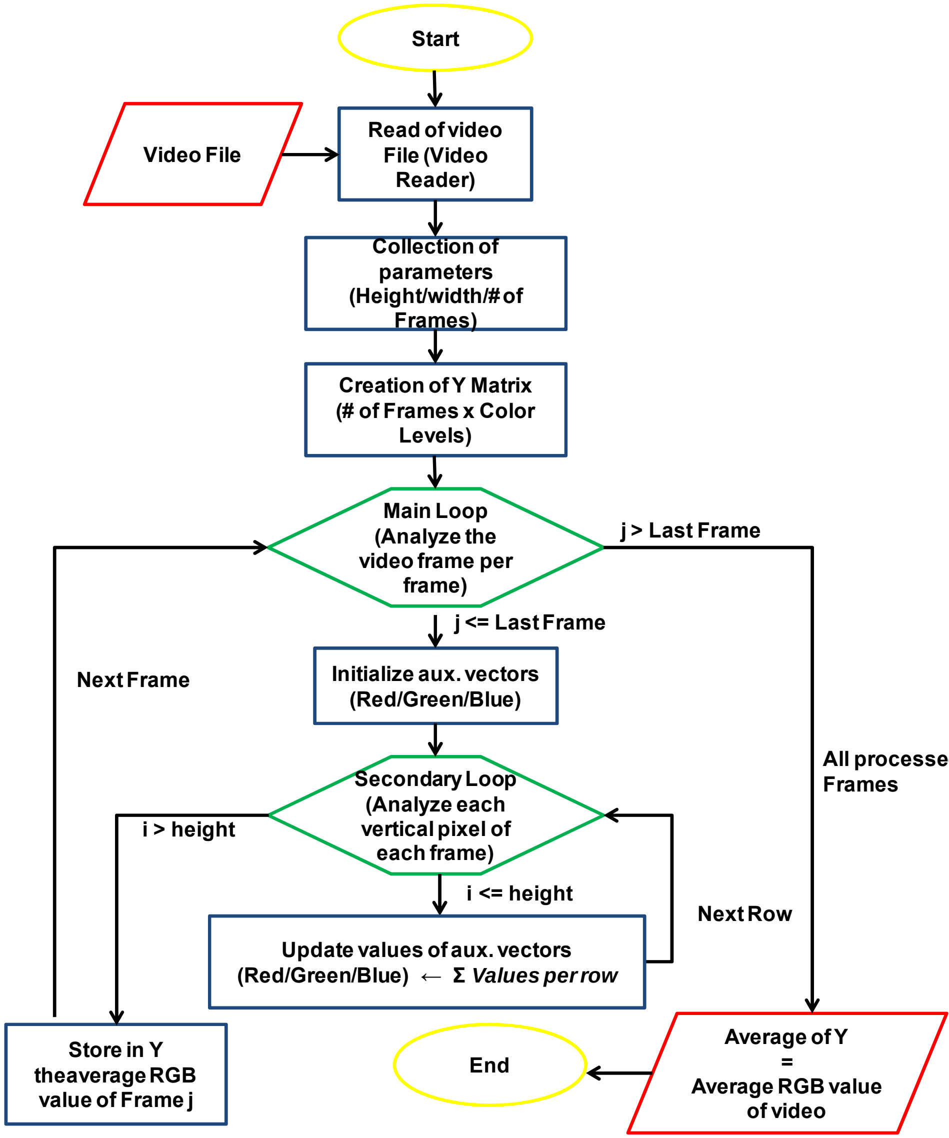


**Figure 3.13.** *Algorithm for determining the predominant color.*

### *3.2.4.2 Network operation algorithm*

In order to serve the requested video, it is important to have available the videos. When a client requests to connect to a surveillance camera, the request is initially received by the server which redirects the request to the cloud that store the video captured by surveillance cameras. Additionally, the server should ask to the client about the limitations.

After connecting with the cloud that stores the videos, the server receives the video and analyzes its features. Finally, the video is sent to the client. Figure 3.14 shows the message exchange between a client and the video server.

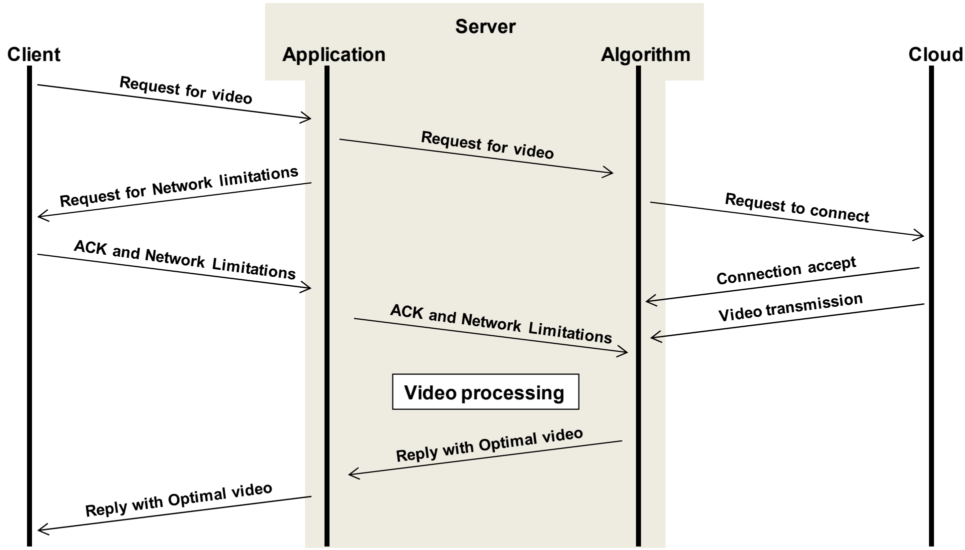


**Figure 3.14.** *Message exchange between a client and the video server.*

Additionally, we need to stablish a connection protocol between the cloud and the server and the server and the client. To carry out these tasks, we have implemented a concurrent TCP socket to establish the communication between both the cloud and the server (see Figure 3.15a) and a UDP socket to establish the communication between both the server and the clients (see Figure 3.15b). As Figure 3.16 shows the server is able to assist several requests from several clients. In the same way, as the server receives a request for a video, the server transmits this request to the cloud in order to download the video. The advantage of using concurrent TCP socket-based protocol is that the cloud or the server can assist several requests, creating for each new request a thread that opens a new process to start with a possible transcoding of a requested video. When the server knows the network requirements, the designed algorithm transcodes the original video according to these requirements and the client does not need to install any additional software. The use of TCP connections between both the cloud and the server ensures a reliable connection with congestion control and flow control mechanisms that guaranties the correct delivery of all data and messages. However, it is recommended to use UDP connections between the server and clients because one of the most important issues is to deliver the requested video as quick as possible.

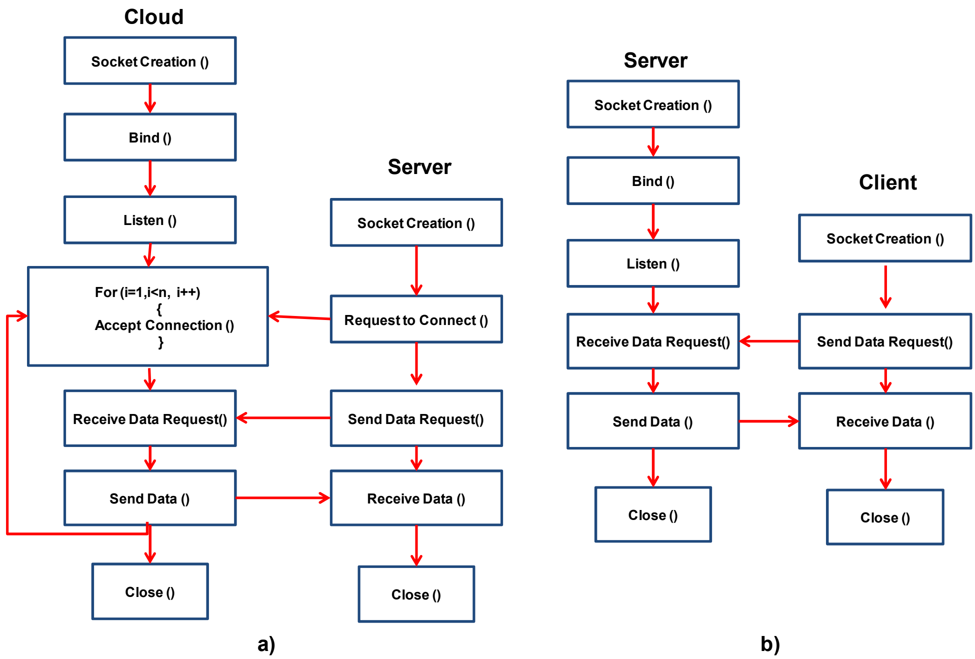


**Figure 3.15.** *Flow diagram of connection between a) the cloud and the server and b) the server and clients.*

### *3.2.4.3 Decision algorithm*

For the realization of the decision algorithm we will take into consideration a previous study [90]. As a summary of this work and design rules for our decision algorithm, we elaborated table 3.2.

**Table 3.2.** *Design rules for our decision algorithm.*

| **Predominant Color** | **Selected Codec as a function of the parameter** | | | | | | |
|---|---|---|---|---|---|---|---|
| | ***Compression size (%)*** | ***Transcoding time*** | ***Subjective Quality*** | ***Consumed Bandwidth*** | ***Maximum Delay*** | ***% of Packet Loss*** | ***Maximum Jitter*** |
| RED GREEN BLUE | H264 | H264/XVID | APCN/H264 | H264 | H264 | H264/XVID | H264 |
| WHITE BLACK | H264 | XVID | APCN/H264 | XVID | XVID | H264/XVID | XVID |

The algorithm, see Figure 3.16, works as follows:

**1** Given a video file to enter the system, first check if there are QoS restrictions.

**2** If not, the video with the original features and codec will be transmitted. If there are such restrictions, it will be checked if there are also quality requirements for user experience.

**3** If this is not the case, it is checked if the video predominates in black or white and if it is negative, XVID will be coded, otherwise it will be recoded in H264.

**4** If there are QoE requirements and the video is predominant black or white, it will be recoded with XVID; if red, green or blue predominate, it will be recoded with H264.

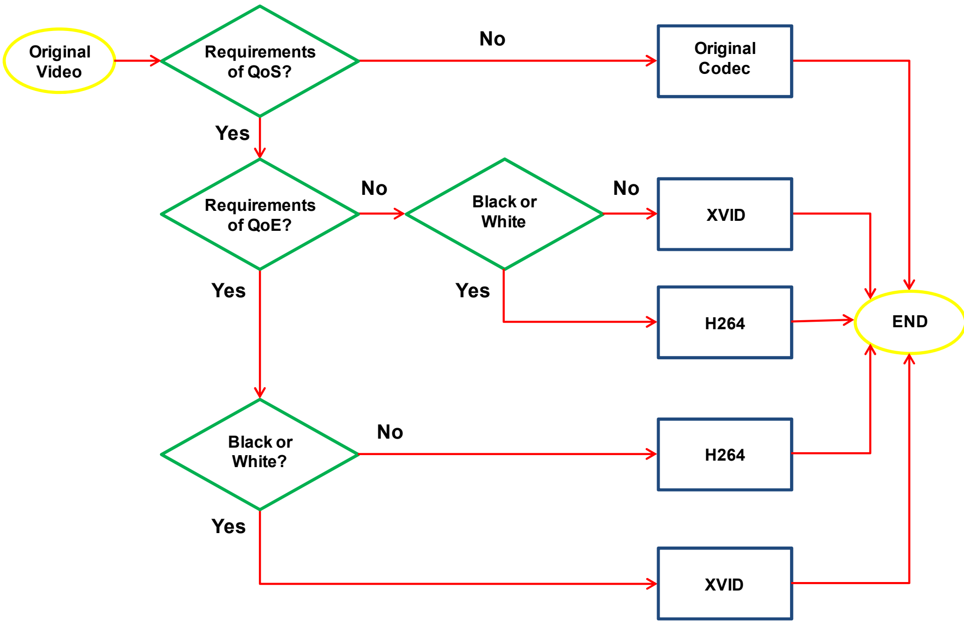


**Figure 3.16.** *Decision algorithm designed for selecting the most suitable compression codec.*

## 3.3 Conclusion

In this chapter we have presented the following proposals:

- A communication protocol and an algorithm that allows the operator to ensure a good IPTV service to its customers by adjusting the quality of the video according to the used device and the bandwidth hired by the customer. These reasons will provide the highest QoE to the customer.

- An algorithm that takes into account the browser type, the compression features, the frame rate of the user's region and the needed video quality.

- A new smart decision algorithm for selecting the best codec for video streams over IP networks. Bearing in mind the QoS considerations provided by the network and QoE of end users, the algorithm takes the information about the color spectrum to select which video codec should be used to save bandwidth.

The dynamic adaptation of the codec used in the video transmission of the same video stream depending on color changes can be translated into a significant network bandwidth saving and improving the QoE perceived by end users. We have also proposed a control protocol for synchronizing the sender and the received when this video delivery is going to take place.

- We have proposed a system based on an algorithm determines the best compression codec for transcoding the video as a function of the measured network parameters and the predominant color of the requested video.

All the proposals presented in this dissertation have the objective to improve the QoE of the final user. Each one of them has been implemented to improve the QoE, as it will be seen in the tests carried out in Chapter 5. When the implementation of all these proposals is carried out jointly, we ensure the improvement of the QoE.

# Chapter 4. Optimization of architecture for videoconferencing

## 4.1 Introduction

As we have already described in the previous sections, when a video communication or videoconference is established, due to the increase of users and the improvements that the devices present in their different evolutions, we foresee that in the near future this form of communication will become a problem for the users, the administrators of the networks, and the service providers. In this chapter, we will propose a new architecture and a new protocol for videoconferencing. Our aim is to reach an optimal E2E QoE, without saturating the network whith the video delivery.

## 4.2 Architecture proposal for videoconference

We must define an E2E QoE Management Scheme for real-time video communication systems, including those operating in resource varying environments.

In our proposal, we must define a human visual perception-based E2E QoE Metric and the methodology of correlating this metric to real-time video data, application/network-level QoS measurements, the capabilities of user devices, and subjective user factors.

We also define network adaptive video encoding and decoding algorithms utilizing device-based E2E QoE-driven feedback and, where available, network-based E2E QoE-driven feedback to achieve real-time adaptation according to the available device and/or network resources.

Besides, we define real-time device-based and network-based feedback control mechanisms that can be used to regulate E2E QoE by one or more of the following methods: application-level objective measurement and reporting of the actual received real-time video signal quality; network-level objective measurement and reporting of the in-transit real-time video signal quality; application-level measurement and reporting of device and/or network resources and QoS performance; and network-level measurement and reporting of device and/or network resources and QoS performance.

To carry out these objectives, we will consider:

- Which parameters affect the QoE.
- The most appropriate algorithms for the network to provide the best QoE.
- The most appropriate algorithms for end-user devices to provide the best QoE.
- How to make the network "adaptive" for this case.
- The best system decision procedure to provide an “adaptive” network

The proposed layer architecture, according to the type of QoS and QoE parameters that are considered, is shown in Figure 4.1.

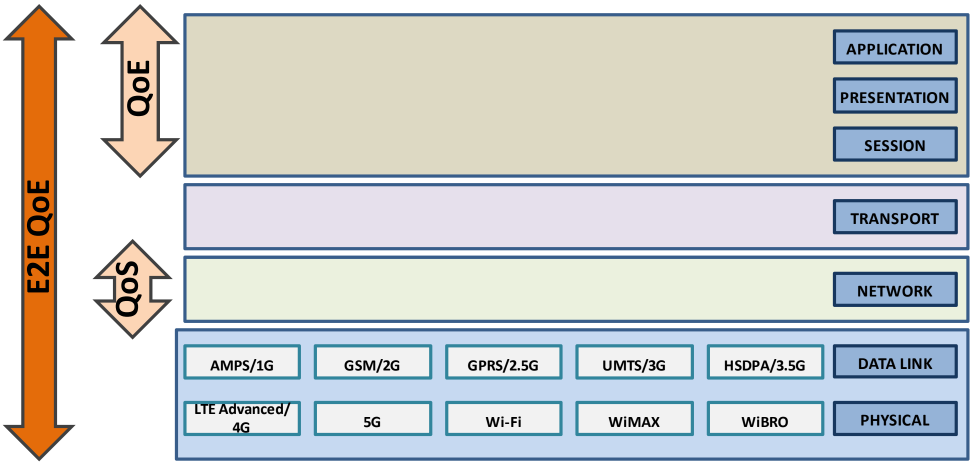


**Figure 4.1.** *Proposed layer architecture, according to the type of QoS and QoE parameters.*

An architecture that includes all the previously established objectives is shown in figure 4.2.

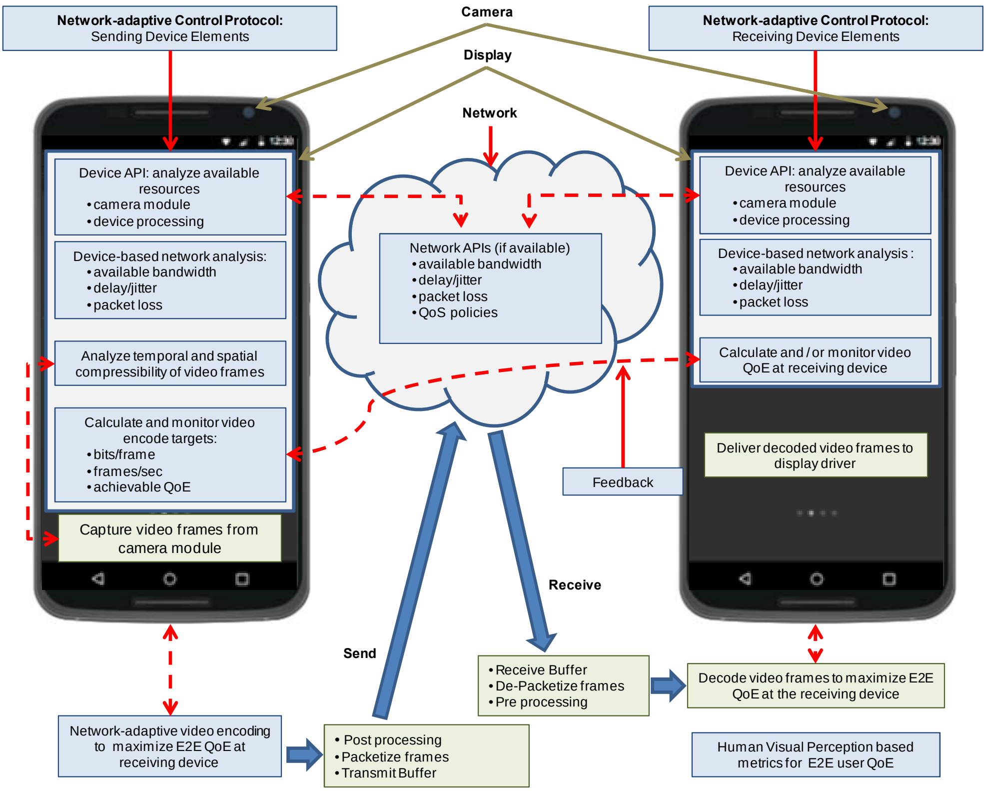


**Figure 4.2.** *Architecture that includes all the established objectives.*

As it can be seen in figure 4.2, our architecture is based on a Network-adaptive Control Protocol. Through this protocol, we manage to adapt the transmission between the end users to the maximum possible QoE. To achieve the goal, we must take into account how to handle a large amount of information, at least during the initial process of the connection.

Information that can be obtained from the source devices:

- Available features of the devices
    - type of camera
    - CPU
    - RAM
    - IOs
- Characteristics of device-based network analysis

  - bandwidth
  - delay
  - jitter
  - packet loss
- Characteristics relative to video compression that can be achieved
  - codecs supported by Software
- Calculate and monitoring video encode targets
  - bits/frame
  - frames/sec
  - achievable QoE

Information that can be obtained from the network between end users:

- Available features of the devices in all the networks where the communication between end users pass through:
  - bandwidth
  - delay
  - jitter
  - packet loss
  - achievable QoE

Information that can be obtained from the destination device:

- Available features of the devices
  - type of camera
  - CPU
  - RAM
  - IOs
- Characteristics of device-based network analysis
  - bandwidth
  - delay
  - jitter
  - packet loss
- Calculate and monitoring video encode targets
  - bits/frame
  - frames/sec

- o achievable QoE

Figure 4.3 shows a generic communication protocol between two users connected to two different service providers, for the establishment of the call. Later, in section 4.5, we will detail the proposed protocol for our architecture.

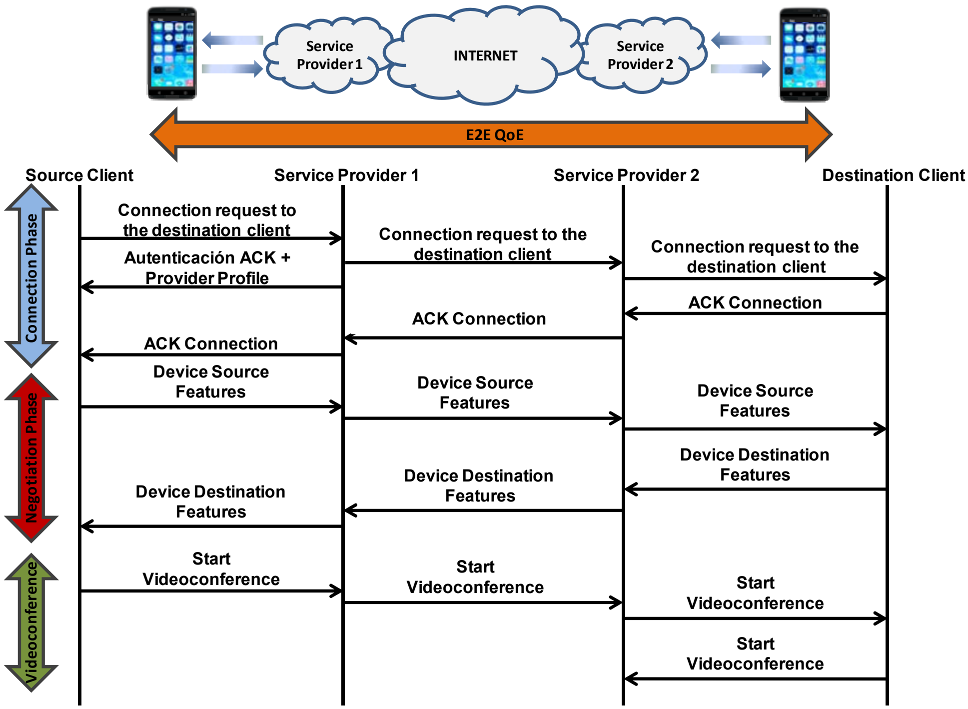


**Figure 4.3.** *Communication protocol between two users connected to two different service providers.*

## 4.3 System Process

In order to design the architecture, we propose three basic processes. They correspond to the basic actions to establish a video communication. Each process is associated to a set of states and transitions that will be detailed later when the system state machine is explained. Figure 4.4 shows the relationship between the processes of the system. Register process is the start and end process of the system. It is the only process that requires the user intervention for executing it.

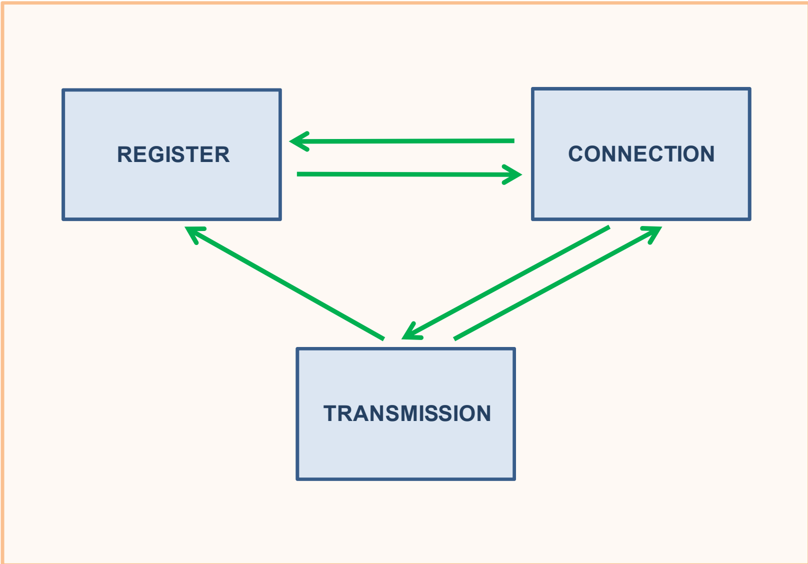


**Figure 4.4.** *System Processes of videoconference architecture.*

System Processes, with the states of each process, are next described in detail:

- **REGISTER PROCESS**: This process includes three states: Idle state, Registered state and Failed state. The user, when starting or ending the video conference, is in the Init state. From the Init state, the user enters the Registered state where the final user will be identified and selected with whom it will establish the communication. The Failed status will be reached whenever video communication is interrupted for any reason. From the Failed state, the user passes to the Init state where they can try again to start a new videoconference.

- **CONNECTION PROCESS**: This process includes two states: Active state and Established state. The Active state is accessed after the registration phase that is when the connection is requested through the application used to connect to the end user. The Active state is where the initial information exchange of the adjustment parameters occurs, which are used by the connected users. The

Active state is also reached, from the Forwarding state, in the case of a small failure during the transmission, trying to recover the transmission again before reaching the Failed state. From the Active state, users can arrive to the Failed state when it is impossible establish the connection with the final user. The Established state is accessed only from the Active state. In this state, videoconference begins. From the Established state, only the Forwarding state can be reached.

- **TRANSMISSION PROCESS**: This process includes only one state: Transmission state. Users arrive at the Forwarding state from the Established state, when users have already begun the video communication. In this state, the instantaneous parameters of the devices and the network are controlled periodically. In case of need, the characteristics of the video communication are varied. In the event of a small communication failure, we can try to re-enter at the Active state, and if the transmission is terminated or it is impossible to establish communication with the end user, it is passed to the Failed state.

## 4.4 Finite State Machine

Figure 4.5 shows the system Finite-State machine. We can see its different states and the transitions between states. In this section, we describe each state of the system and the conditions and events that will make the node change from one state to another inside a process.

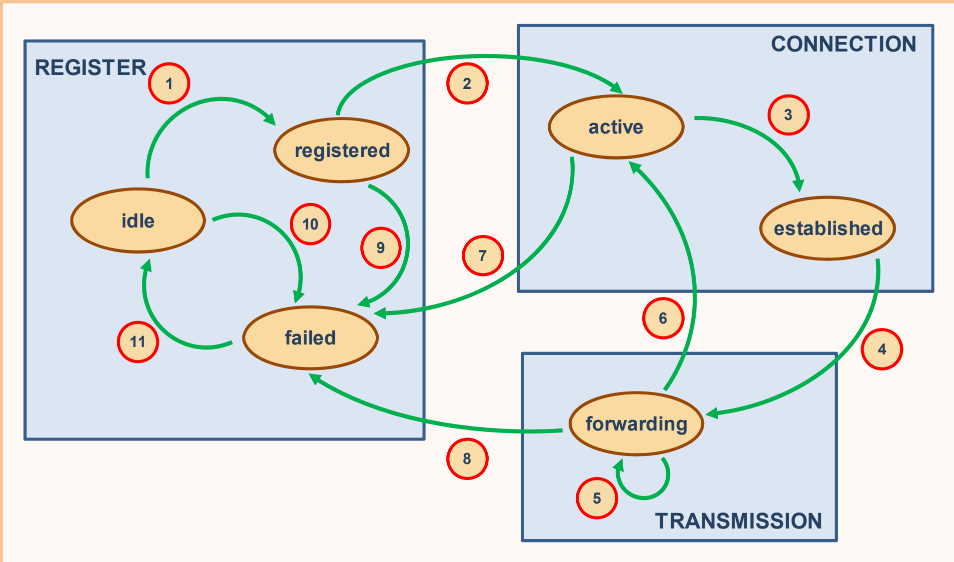


**Figure 4.5.** *System Finite-State machine.*

The processes included in figure 4.5 are the following ones:

- **Idle state.** It is the state in which the user is, before initiating access to the application to establish the videoconference or once the videoconference is finished. The user, once the application is selected to make a videoconference, will go from this state to the Registered state.
- **Registered state.** Registered state is accessed only from the Idle state. The user initiating the videoconference, depending on the employed software, must initiate the authentication process in the server. Once authenticated, it will search for the remote user that it wants to connect to, in its own database or in the server database. Once the end user is found, it will demand the connection with the selected end user to the server. The server tries to make contact between the users to establish an initial connection, and they will go to the Active state. In case the user that is initiating the call does not want to connect with any of the available users or cannot establish a connection with the end user, it will go to the Failed state.
- **Active state.** The Active state can be reached from the Registered or Forwarding states. From the Active state, we can move to Established

or Failed states. Once the initial contact between the users participating in the videoconference has been established, in the Active state, an exchange of parameters between the end devices of the users will be initiated, at the same time that the information of the network parameters is obtained. Thanks to the information obtained, an algorithm that will allow reaching an agreement to reach the maximum E2E QoE among the users at that moment will be applied. From this moment on, it will go to Established status. In case that one of the two users rejects or terminates the connection, or a connection agreement cannot be reached due to the parameters of any of the user devices or of the network, it will go to the Failed state.

- **Established state.** Established status can only be reached from the Active state. From the Established state you can only move to the Forwarding state. Once the Established state is reached, the video starts from the devices of the connected users, moving to the Forwarding state.
- **Forwarding state.** The Forwarding state can be reached from the Established state and from the Forwarding state itself. Once the video conference starts, we will remain in the Forwarding state while everything is working correctly. In this state, the final devices will continuously control the characteristics of the devices themselves and the network, so that when any variation appears, the appropriate measures are taken and the maximum E2E QoE is still maintained. We can vary the codec that was used until then, in case of need for more compression. Periodically, Ack will be exchanged (both for an appropriate videoconference reception and for the adjustment of parameters) between the devices of the users participating in the videoconference. It will establish a maximum time period (time out) that, if exceeded, the corresponding Ack will not be received. Thus, it will be considered that the videoconference is failing. If a failure occurs, it will go back to the Active state to try to re-negotiate the parameters of the devices and the network parameters and go on to re-launch the transmission. In case of not being able to get to establish the connection again, we will go to the Failed state. In the event that any of the users ends the videoconference, we will go to the Failed state.

- **Failed state.** The Failed status is reached if the videoconference did not work correctly or one of the users decided to disconnect. The Failed state is reached from the Idle, Registered, Active and Forwarding states. From the Failed state, it passes to the Idle state to start the whole process again.

# 4.5 Protocol proposal for videoconference

Figure 4.6 shows the protocol proposed for the start of the establishment of the connection. It includes the generic actions that will be carried out during the Idle and Registered states.

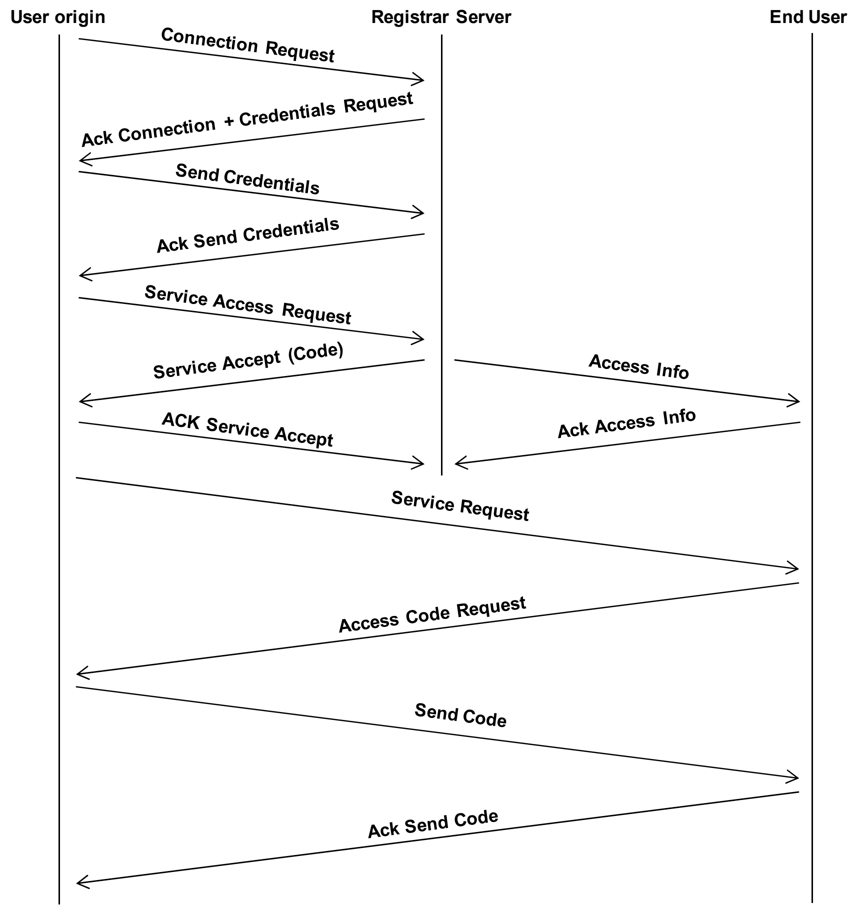

**Figure 4.6.** *Protocol proposed for Idle and Registered states.*

Figure 4.7 shows the proposed protocol for the Active state. In this state, users exchange characteristic parameters of their devices and also of the network, in order to achieve the transmission of the videoconference with the maximum E2E QoE.

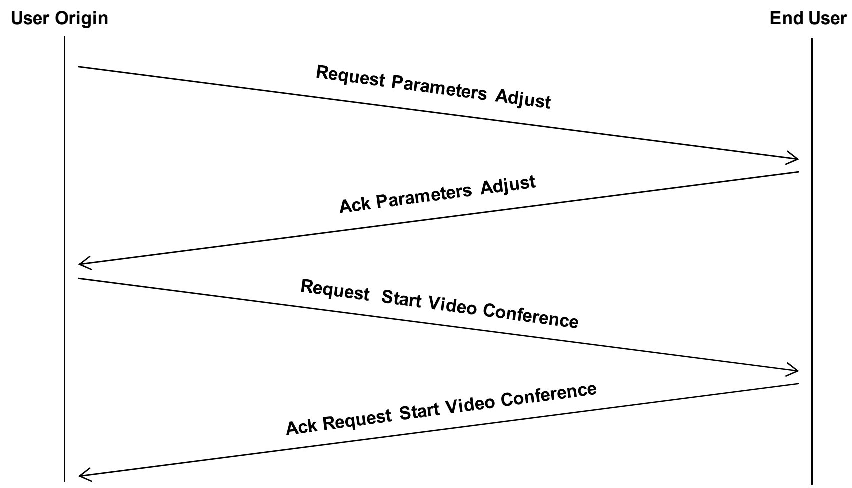


**Figure 4.7.** *Protocol proposed for Active State.*

Figure 4.8 shows the proposed protocol for the Established state. In this state, the video and audio transmission of the videoconference between the interlocutors begins. The transmission of audio and video will be done in both directions simultaneously, although in Figure 4.8 the delivery can be observed in only one direction.

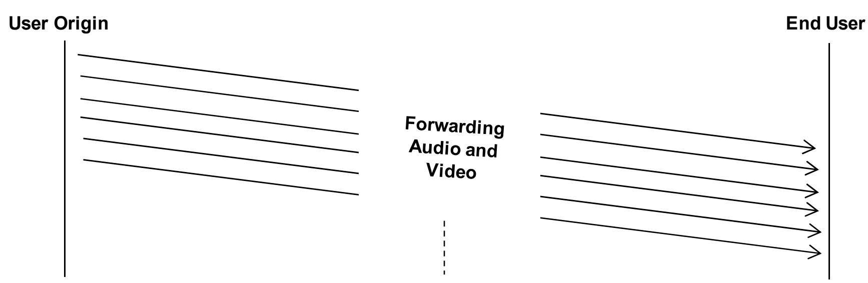


**Figure 4.8.** *Protocol proposed for Established State.*

Figure 4.9 shows the proposed protocol for the Forwarding state, but only during the correct operation of the video conference. It can be seen that the transmission initiated during the Established state continues. Its correct operation is being controlled by the exchange of Ack's. They are received before the Time Out expires. The transmission of audio, video and Ack's will be done in both directions simultaneously, although Figure 4.9 shows the transmission in only one direction.

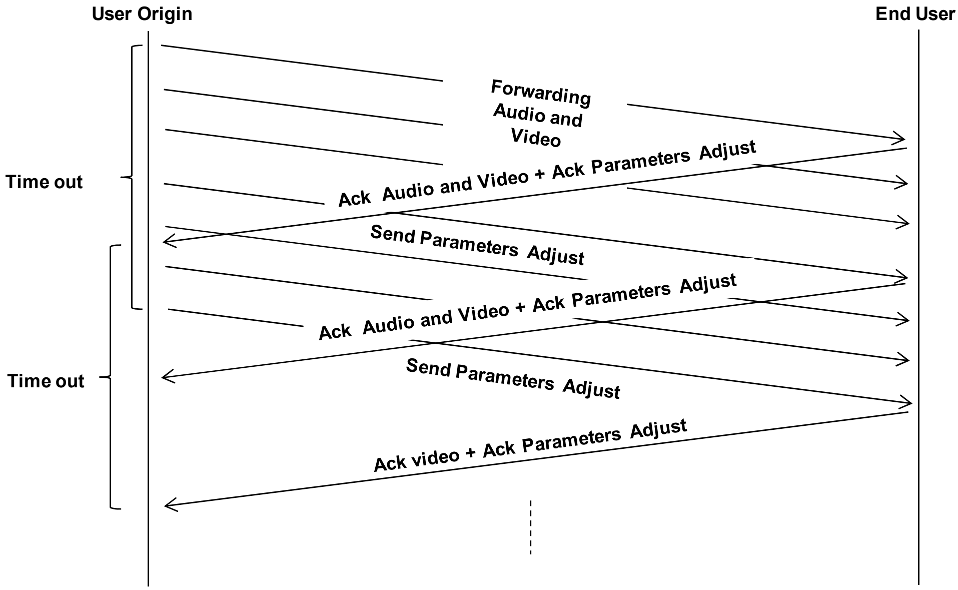


**Figure 4.9.** *Protocol proposed for Forwarding State when run correctly.*

Figure 4.10 shows the proposed protocol for the Forwarding state, but when it stops working correctly, since the Ack is not received from the remote user within the Time Out. When this situation occurs, it will go back to the Active state to try to recover the transmission. The transmission of audio and video will be done in both directions simultaneously, although in Figure 4.10 the transmission can be observed in only one direction.

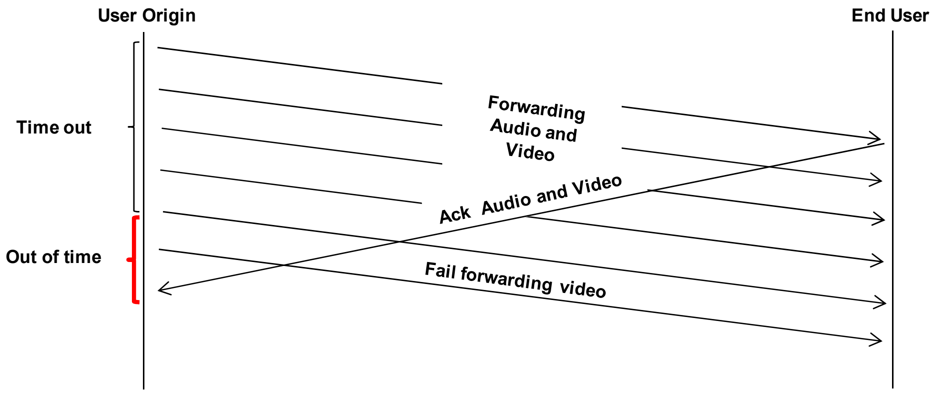


**Figure 4.10.** *Protocol proposed for Forwarding state when there is a problem.*

Figure 4.11 shows the proposed protocol for the transition from the Active state to the Fail state. As seen in Figure 4.11, the transition can occur for two reasons, after several requests for adjustment parameters that have not been answered or after several failures in the attempt to start the video conference. The transmission of audio and video will be done in both directions simultaneously, although in Figure 4.11 the transmission can be observed in only one direction.

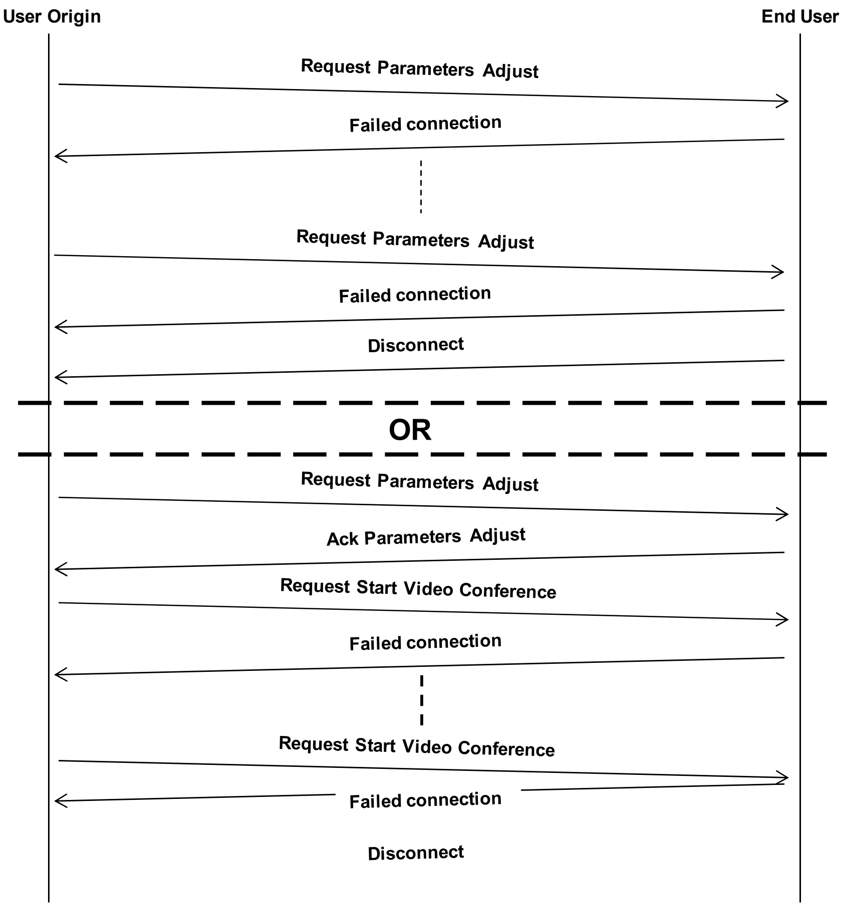


**Figure 4.11.** *Protocol proposed for the transition from the Active to the Fail state.*

Figure 4.12 shows the proposed protocol for the transition from the Forwarding state to the Fail state. As shown in Figure 4.12, the transition occurs when one of the users involved in the videoconference decides to end it, without any transmission error. The transmission of audio and video will be done in both directions simultaneously, although in Figure 4.12 the transmission can be observed in only one direction.

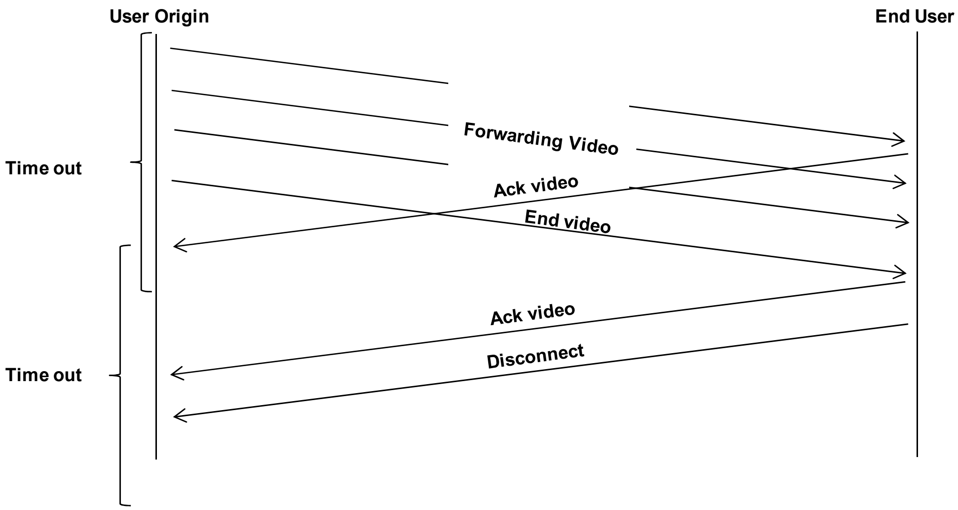


**Figure 4.12.** *Protocol proposed for the transition from the Forwarding to the Fail state.*

Figure 4.13 shows the proposed protocol for the transition from the Registered state to the Fail state. As seen Figure 4.13, the transition occurs when, once the User Origin is authenticated, the connection to the End User can not be established. When this connection attempt fails, it passes to the Fail state in which it will be disconnected.

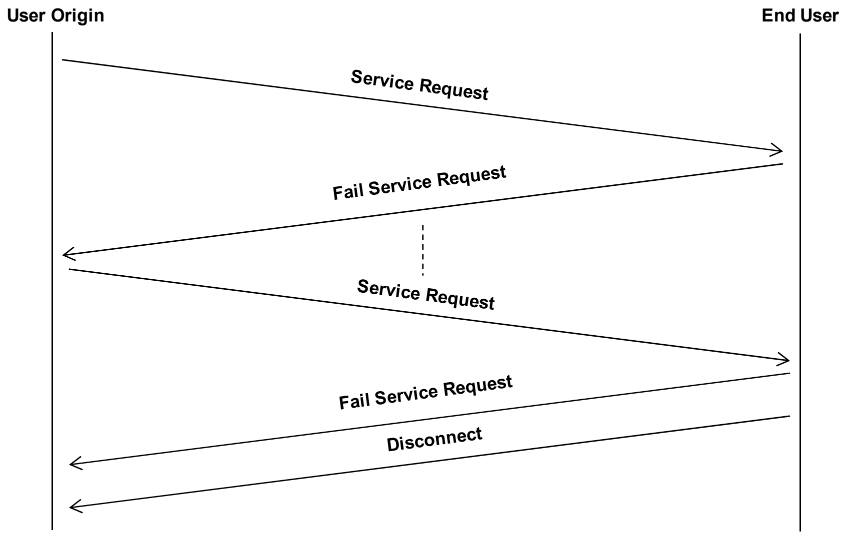


**Figure 4.13.** *Protocol proposed for the transition from the Registered to the Fail state.*

Figure 4.14 shows the proposed protocol for the transition from the Idle state to the Fail state. As seen in the image, the transition occurs when, once the videoconferencing software application starts, the User Origin cannot be authenticated in the server. When this authentication attempt fails, it passes to the Fail state in which it will be disconnected.

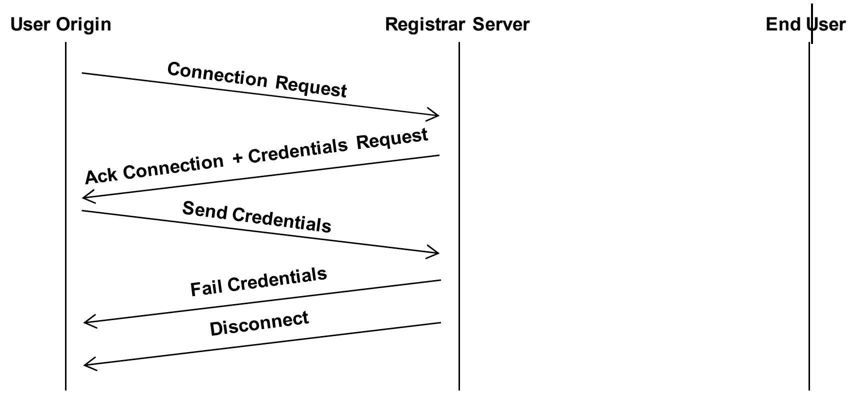


**Figure 4.14.** *Protocol proposed for the transition from the Idle to the Fail state.*

Finally, Figure 4.15 shows the proposed protocol for the transition from the Fail state to the Idle state. As shown in Figure 4.15, the transition occurs when the credentials sent by the User Origin failed and, when it cannot be authenticated to the server. It sends it to the Idle state, where it can try to start a new connection.

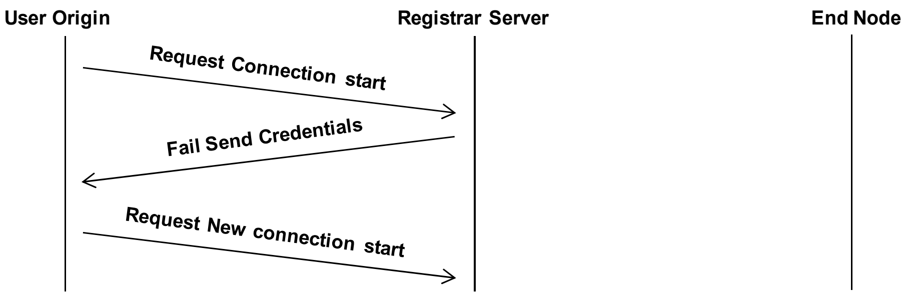


**Figure 4.15.** *Protocol proposed for the transition from the Fail to the Idle state.*

# 4.6 Conclusion

In this chapter, we have presented a new architecture and a new protocol to optimize videoconference.

First, we have defined an E2E QoE Management Scheme. This scheme utilizes correlation of both subjective and objective E2E QoE with received real-time video data (stream header and/or video signal), application-level QoS measurements, and network-level QoS measurements.

We define real-time device-based and network-based feedback control mechanisms that can be used to regulate E2E QoE, and we present our proposal of architecture for videoconference.

We propose three basic processes, which correspond to the basic actions to establish a videoconference (Register, Connection and Transmission).

Later, we propose a Finite-State Machine, and we present and define the different states.

Finally, we present our new protocol for videoconference.

# Chapter 5. Performance Test

# 5.1 Introduction

The evolution of multimedia delivery has been possible thanks to the development of control mechanisms to promote QoS and QoE in IP networks: traffic prioritization, streaming protocols, etc. [157]. They have brought benefits to both service operators and users in many aspects such as: obtaining an efficient communication, optimizing resources and reducing costs in both, the part of the service providers and in consequence to the users, optimizing network administration, delivery of new services, and efficient and secure network infrastructure developing. New technologies such as adaptive streaming [96] and Software Defined Networks [158] are providing research advances in this topic.

A QoS categorized model of user-centric let us know the performance quality goals of audio, data and video. These goals are detailed in the Recommendation UIT-T G.1010 [159] and Recommendation UIT-R M.1079-2 [160]. So, we primarily must take into account parameters such as latency, jitter, and bandwidth and packet loss in both wired and wireless networks [102][135]. Others parameters we are interested in are shown in Cisco VNI report [161]. Some of them are internet user demand, network connections growth, as well as the generation of internet video minutes per month in a global scale and especially the number of video users in Internet.

In this chapter we will present the different tests we have done over the years to study the parameters that affect the transmission of video and video conferencing.

# 5.2 Multimedia over SDN

In this study, different tests have been carried out when a video is sent from an IPTV server through the network. The study has been divided into: coding phase, transmission phase and evaluation phase.

We perform a comparative study to analyze the accuracy in terms of performance of a real network and a network implemented with Mininet. To do this, we have tested the transmission of multimedia streaming using both real and virtualized devices. Transmissions have been performed over UDP protocol, using the Real-Time Transport Protocol (RTP) [162]. The virtualization of network devices and hosts has been carried out with Mininet [163]. Mininet is a highly custom flexible SDN emulator that supports the OpenFlow protocol. Our goal is to compare the results between real and virtualized devices, to verify the accuracy of the results of multimedia delivery in virtualized environments.

We are going to introduce the SDN emulator and the real network topology used in our test bench.

### 5.2.1 Devices and equipment

This subsection describes the devices and equipment used to perform our study.

The real topology is composed by the following equipment:

- 2 Router Cisco 2811 Series [164], that runs an IOS C2800NM-ADVIPSERVICESK9-M, Version 15.0 (1) M, Release Software (fc2). It has 2 Fast Ethernet and 2 Serial (sync/async) interfaces and 64 MBytes of Compact Flash;
- 2 Switches Cisco Catalyst WS-C3560-24PS-E [165] that runs an IOS C3560-IPSERVICESK9-M,Versión 12.2 (53) SE2, Release Software (fc3). It has 24 Fast Ethernet and 2 Gigabit Ethernet interfaces and 16 Mbytes of Flash memory;
- 1 Desktop PC that has an Intel Core Quad Q9400 CPU @2.66 Ghz processor, 6 Gb of RAM memory, 1 Network Interface Card (NIC) Intel 82579V Gigabit Ethernet and Windows 7 Professional - 64 bits Operative System;
- 1 Desktop PC that has an Intel Core i5-2400 CPU @3.10 Ghz, 4 Gb RAM memory, 1 NIC Intel 82579V Gigabit Ethernet and Windows 7 Enterprise - 64 bits as Operating System.

In order to connect the PCs and network devices, we have used:

- Cables UTP Cat.5e to connect the NICs (1.5 meters);

- Cisco CAB-SS-V35MT Smart Serial DTE/DCE WAN Cable serial cable to connect WAN interfaces of routers.

To design and develop the virtualized topology we have used a Laptop composed by an Intel i7-4500UCPU @ 2.70 Ghz processor, 16 Gb RAM memory, 1 10/100/1000 Mbit/s NIC, and Ubuntu 14.04 - 64 bits as Operating System.

### 5.2.2 Software used

In addition to the operating systems, we have used some software applications. We used, on the one hand, a software application to send and receive multimedia streaming and, on the other hand a software application to capture the received packets on the target PC.

In the real topology, VLC [166], version 2.0.8 Twoflower has been used to stream video from the server and in the client to receive the video streaming. To capture and analyze the received traffic, we have used Wireshark [167], version 1.10.5.

In the virtualized topology, we have used VLC version 2.16 for both, send and receive video streams and Wireshark version 1.10.6 to gather data.

### 5.2.3 Video and multimedia streaming characteristics

We have used a video with the following characteristics: 50 seconds length, 800X600 frame size, 1026 Kbps bit rate (BRV) and 60 frames per second.

To send the multimedia traffic, we have used a Real-Time Transport Protocol (RTP) specified by the IETF through the RFC 1889 [162]. This was designed by the IETF's Audio-Video Transport Working Group to support video conferences with multiple geographically dispersed participants. RTP, by itself, does not guarantee the real-time delivery of multimedia data (since this is dependent on network characteristics); however, it provides the wherewithal to manage the data when it arrives. RTP is located above the UDP transport. The main function of RTP is to implement the sequence numbers of IP packets to reset the voice or video even when the underlying network changes the order of the packets.

The RTP header has a minimum size of 12 bytes. After the header, an optional header extension may be present. This is followed by the RTP payload, the format of which is determined by the particular class of application. Some underlying protocols may require a RTP encapsulation to be defined. Typically, one packet of the underlying protocol contains a single RTP packet, but several RTP packets may be contained if permitted by the encapsulation method. The main information of a RTP packet is the sequence number, timestamp and unique identifier for the source (SSRC).

In our study we send the RTP traffic through the UDP-port 5004. We need to configure this port in both PCs, source and destination, to properly work the multimedia streaming.

### 5.2.4 Physical topology

We have established six different cases in three topologies, for both real and virtual scenarios. This subsection describes the physical topologies we have used along the study when we work with the real equipment.

The first topology consists of two desktop PCs connected by a crossover cable, as it is shown in Figure 5.1. In this case we have sent traffic at 1 Gbps.

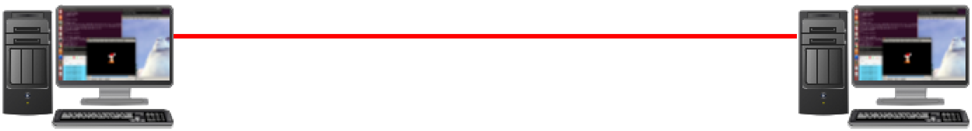

**Figure 5.1.** *Physical Topology 1.*

The second topology consists of two desktop PCs connected, by straight-through cable, using 2 switches that are connected by a crossover cable, as it is shown in Figure 5.2. The data transfer rates used in this case has been 10 Mbps and 100 Mbps.

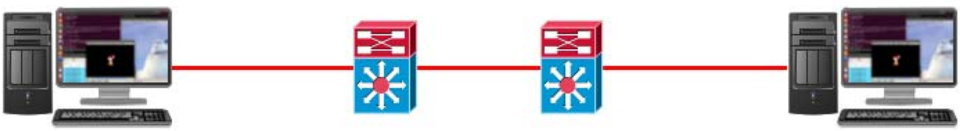

**Figure 5.2.** *Physical Topology 2.*

As Figure 5.3 shows, the third topology has added two routers between the switches. In this case, the link between routers has been a serial cable. The traffic has been sent at 2 Mbps, 4 Mbps and 8 Mbps through the WAN interfaces of the routers. The data transfer rates between PCs and switches is 100 Mbps.

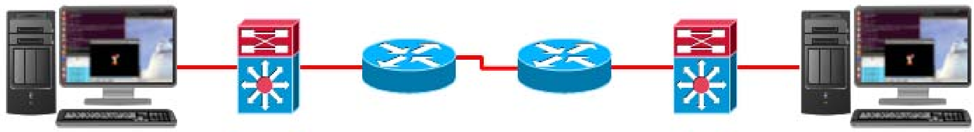

**Figure 5.3.** *Physical Topology 3.*

In the virtualized networks, we have used a PC with Mininet, in which we reproduced the same topologies that were employed with real equipment, as it is shown in Figure 5.4.

**Figure 5.4.** *PC with virtualization used in all topologies.*

### 5.2.5 SDN emulator: Mininet

There are several network simulators and emulators, but the most frequently used Simulators and Emulators for SDNs are the following ones:

- NS3 [168]: It is a C++ library which provides a set of network simulation models implemented as C++ objects and wrapped through python. It has OpenFlow support built in to emulate an Openflow environment and it can also be used for real-time simulations.
- EstiNet 9.0 [169]: It is a simulator and emulator can simulate thousands of Ver 1.3.4 and Ver 1.0.0 OpenFlow switches and run the real-world OpenDaylight [170], Ryu [171], NOX, and Floodlight [172] controllers without any modification to control these switches during simulation. Its

performance simulation results are realistic, accurate, and repeatable. It supports both of the simulation mode and the emulation mode.

- Mininet: It is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking. Version 2.2.1, installed natively on Ubuntu 14.04, can be used as a rapid prototyping for SDN. Mininet enables researchers to quickly create, interact with, customize and share a software defined network prototype, and provides a smooth path to running on hardware.

With Mininet, we can create a realistic virtual network, running real kernel, Switch and application code, on a single machine. The machine can be a Virtual Machine, or a machine virtualized through the cloud or native. For our study we have used Mininet version 2.2.1, with a native installation on Ubuntu 14.

Network topologies can be created using the commands: self.addHost( ) for adding a host, self.addSwitch( ) for adding a switch, self.addNode( ) for addinng a Linux router, and self.addLink( ) to establish links between hosts, switches and routers. To change the link speed, we use the optional parameter ‘bw=xx’, where xx is the speed in Mbps. An example of the code to set up a network topology with 2 PCs and 2 switches with default link speed is shown in Figure 5.5.

```
# Add hosts and switches
h1 = self.addHost( 'h1' )
h2 = self.addHost( 'h2' )
s1 = self.addSwitch( 's1' )
s2 = self.addSwitch( 's2' )
# Add links
self.addLink( h1, s1 )
self.addLink( h2, s2 )
self.addLink( s1, s2 )
```

**Figure 5.5.** *Example of code to add hosts and switches in Mininet.*

Figure 5.6 shows the result of defining the topology previously defined, with all links at 100 Mbps. Links between hosts and switches are also shown in the Mininet command line.

```
oscar@sony: ~
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(100.00Mbit 0ms delay 0% loss) (100.00Mbit 0ms delay 0% loss) (h1, s1) (100.00Mbi
t 0ms delay 0% loss) (100.00Mbit 0ms delay 0% loss) (h2, s2) (100.00Mbit 0ms dela
y 0% loss) (100.00Mbit 0ms delay 0% loss) (s1, s2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(100.00Mbit 0ms delay 0% loss) (100.00Mbit 0ms delay 0% loss) (100.00Mbi
t 0ms delay 0% loss) (100.00Mbit 0ms delay 0% loss)
*** Starting CLI:
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s2-eth1 (OK OK)
s1-eth2<->s2-eth2 (OK OK)
mininet> xterm h1 h2
mininet>
```

**Figure 5.6.** *Launch topology.*

Figure 5.7 shows the links for topology formed with 2 routers and 2 switches (scenario of topology 2).

```
oscar@sony: ~
mininet> links
h1-eth0<->s1-eth2 (OK OK)
h2-eth0<->s2-eth2 (OK OK)
r1-eth1<->r2-eth1 (OK OK)
s1-eth1<->r1-eth2 (OK OK)
s2-eth1<->r2-eth2 (OK OK)
mininet>
```

**Figure 5.7.** *Links, topology with 2 routers and 2 switches.*

After defining the network topology within Mininet, next step is to send a video from host h1 to host h2. To this aim, we have used VLC version 2.1.6 for the video streaming in both, sender h1 and receiver h2. Thus, an Xterm session is launched from Mininet CLI for each host as shown in the Figure 5.8. In addition, we launch another Xterm session in the receiver in order to capture the received packets with Wireshark version 1.10.6. From this capture, we analyze the parameters of the video streaming for each situation. Figure 5.8 shows the desktop of a laptop when it is running Mininet with the network topology 2 and streaming the video from host h1 to host h2. The video is only displayed in the receiver and Wireshark captures the frames received in the network interface of host h2.

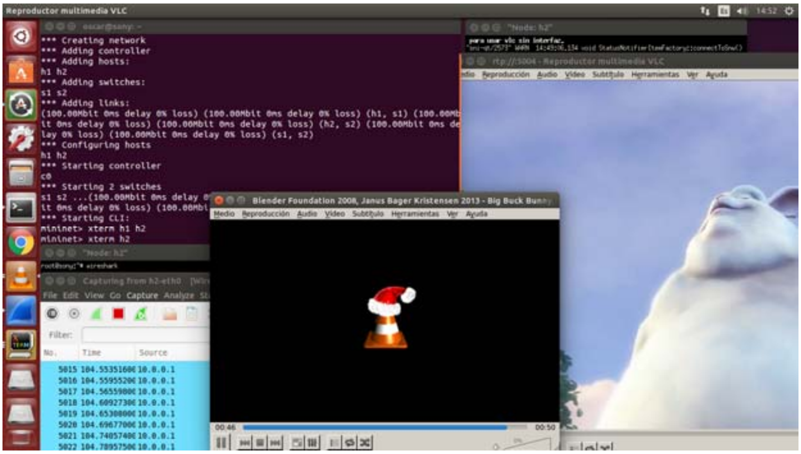


**Figure 5.8.** *Screen capture where Mininet is running.*

## 5.2.6 Measurements and discussion

We show the results obtained in both cases, when multimedia streaming is being delivered over the network and in the virtual topology using Mininet.

### 5.2.6.1 Bandwidth

We present the results related of bandwidth obtained for the three topologies.

### 5.2.6.2 Bandwidth consumed in topology 1

In Figure 5.9 we can see the bandwidth consumption values of the real topology and the values obtained in the virtual topology. The data have similar values for both topologies. The mean value of bandwidth in real topology is 1337.26 Kbps while for virtual topology is 1341.95 Kbps. The maximum values for real topology and virtual topology are the same 2592.67 Kbps. We obtained different minimum values, 162.72 Kbps for real topology and 86.78 Kbps for virtual topology (the initial 20 packets are not taken into account because the streaming was initiating). The variance of the data is 247128.43 for real topology and 244664.70 for virtual topology. The standard deviation is 497.12 in real topology and 494.63 in virtual topology. The data does not follow a normal distribution neither in real nor in virtual topology.

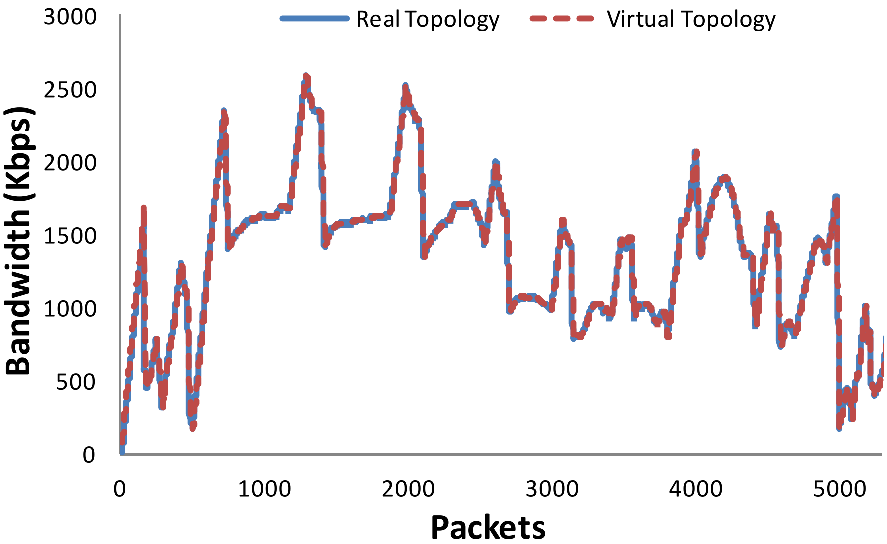


**Figure 5.9.** *Bandwidth in topology 1, link at 1 Gbps.*

#### *5.2.6.3 Bandwidth consumed in topology 2*

When we study the bandwidth consumption in topology 2, we can observe two different cases, when the link between switches is established at 10 Mbps and at 100 Mbps.

In Figure 5.10 we can see compared the values of bandwidth consumed in the real topology with the values of the virtual topology, when the link between switches is at 10 Mbps. The data have similar values in both topologies. The mean value of the bandwidth consumed in real topology is 1341.77 Kbps while for the virtual topology is 1351.85 Kbps. The maximum value for real topology and virtual topology are the same 2592.67 Kbps, but the minimum is slightly different, 162.72 Kbps for real topology and 173.57 Kbps for virtual topology (the initial 20 packets are not taken into account because the streaming was initiating). The variance of the data is 248874.16 for the real topology and 248150.91 for the virtual topology. The standard deviation is 498.87 in the real topology and 498.14 in the virtual topology. The data does not follow a normal distribution neither in real nor in virtual topology.

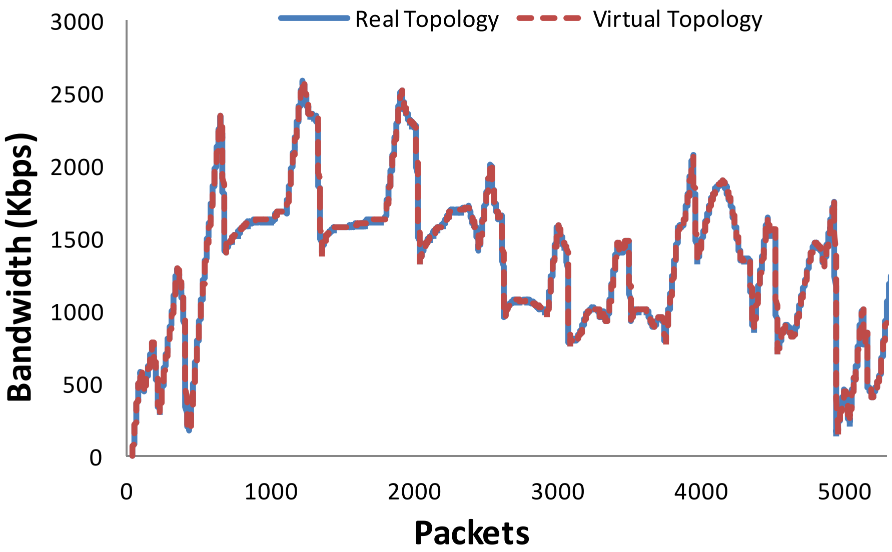


**Figure 5.10.** *Bandwidth in topology 2, link at 10 Mbps.*

Figure 5.11 shows the comparison between the values of the bandwidth consumed in the real topology and in the virtual topology, when the link between switches has 100 Mbps. The data presents similar value for both topologies. The mean value of bandwidth in real topology is 1340.22 Kbps while for virtual topology is 1350.78 Kbps. The maximum value for both real and virtual topologies is the same 2592.67 Kbps. The minimum value has been slightly different, 162.72 for real topology and 173.57 for virtual topology (for the minimum, the initial 20 packets are not taken into account because the streaming was initiating). The variance of the data is 249230.523 for real topology and 248179.04 for virtual topology. The standard deviation is 499.22 in real topology and 498.17 in virtual topology. The data does not follow a normal distribution neither in real nor in virtual topology.

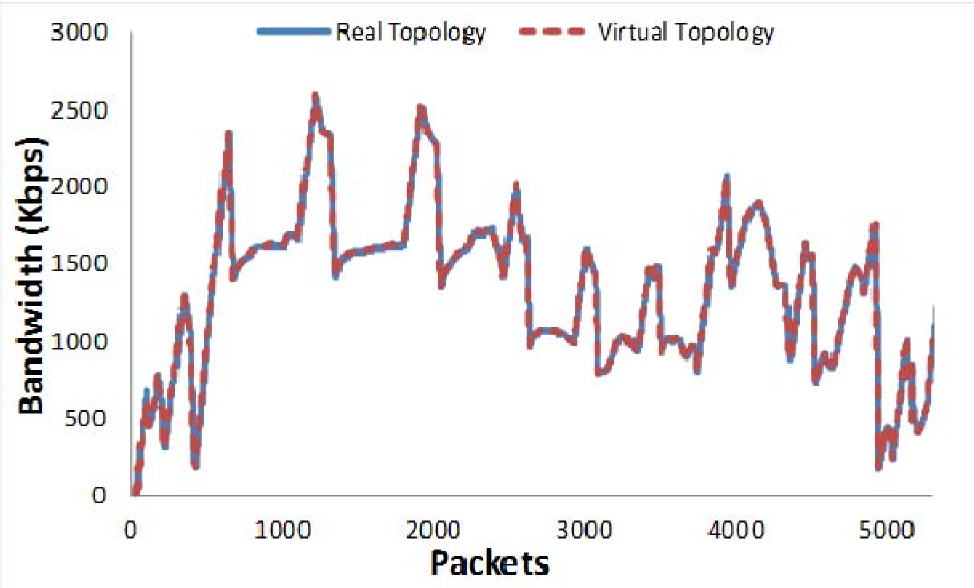


**Figure 5.11.** *Bandwidth in topology 2, link at 100 Mbps.*

#### *5.2.6.4 Bandwidth consumed in topology 3*

When we study delay in topology 3, we can observe three different cases, when the WAN link between routers is established at 2, 4 and 8 Mbps.

Figure 5.12 shows the consumed bandwidth values of the real topology and the virtual topology, when the WAN link between routers is configured at 2 Mbps. Both topologies have similar data. The mean consumed bandwidth in real topology is 1335.49 Kbps while for virtual topology is 1372.51 Kbps. The maximum value for real topology is 1963.49 Kbps, while for virtual topology is 1996.03. The minimum value has been the same in both cases, 173.57 Kbps (for the minimum, the initial 20 packets are not taken into account because the streaming was initiating). The variance of the data is 218658.95 for real topology and 225180.89 for virtual topology. The standard deviation is 467.60 in real topology and 474.53 in virtual topology. The data does not follow a normal distribution neither in real nor virtual topology.

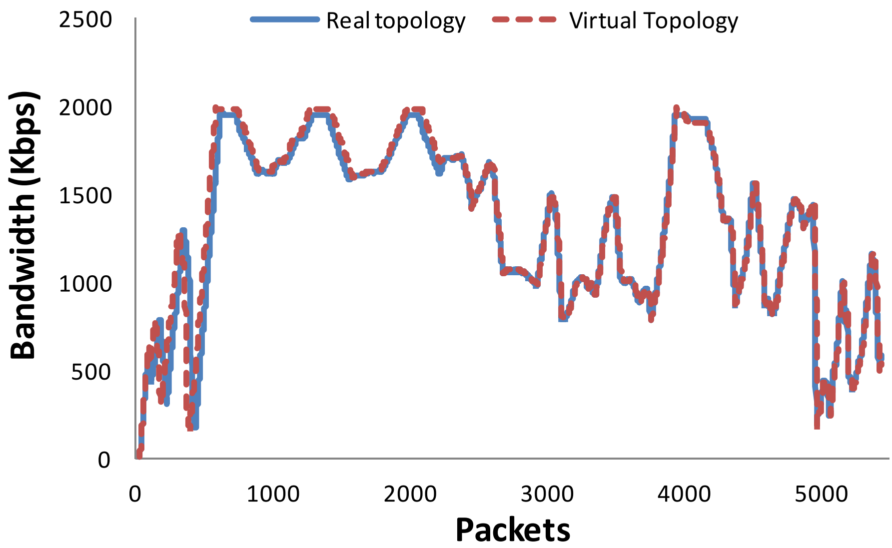


**Figure 5.12.** *Bandwidth in topology 3, WAN link at 2 Mbps.*

In Figure 5.13 we can observe the consumed bandwidth for the real topology and the virtual topology, when the WAN link between routers is configured at 4 Mbps. The data presents similar value in both topologies. The mean value of bandwidth in the real topology is 1343.47 Kbps while for the virtual topology is 1353.89 Kbps. The minimum and maximum values for real topology and virtual topology are the same, 162.72 Kbps and 2505.89 Kbps respectively (for the minimum, the initial 20 packets are not taken into account because the streaming was initiating). The variance of the data is 245943.93 for real topology and 244590.52 for virtual topology. The standard deviation is 495.92 in real topology and 494.56 in virtual topology. The data does not follow a normal distribution neither in real nor virtual topology.

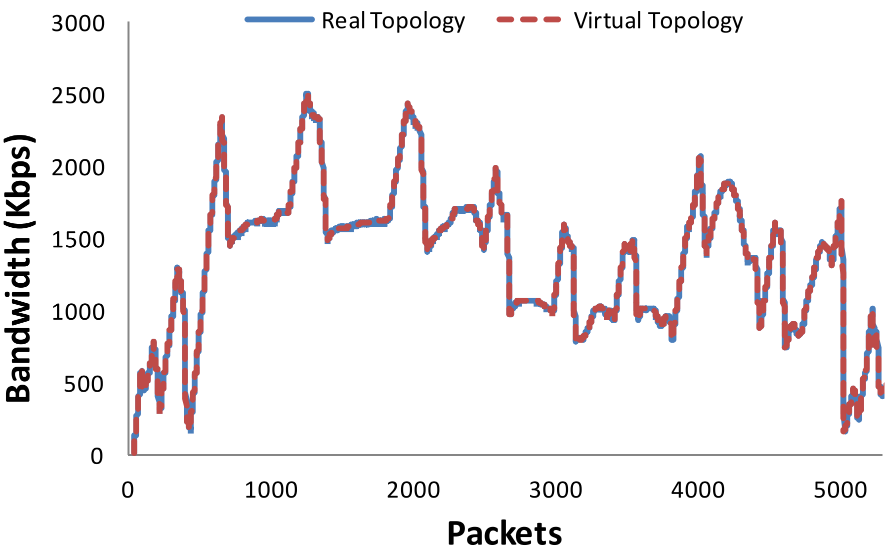


**Figure 5.13.** *Bandwidth in topology 3, WAN link at 4 Mbps.*

Figure 5.14 shows the values of the bandwidth consumed by the real topology compared with the values of virtual topology, when the WAN link between routers is consumed at 8 Mbps. The data presents similar value for both topologies. The mean value of bandwidth in real topology is 1341.80 Kbps while for virtual topology is 1352.85 Kbps. The minimum and maximum values for real topology and virtual topology are the same, 162.72 and 2592.67 Kbps respectively (for the minimum, the initial 20 packets are not taken into account because the streaming was initiating). The variance of the data is 248882.63 for real topology and 247659.12 for virtual topology. The standard deviation is 498.88 in real topology and 497.65 in virtual topology. The data does not follow a normal distribution neither in real nor virtual topology.

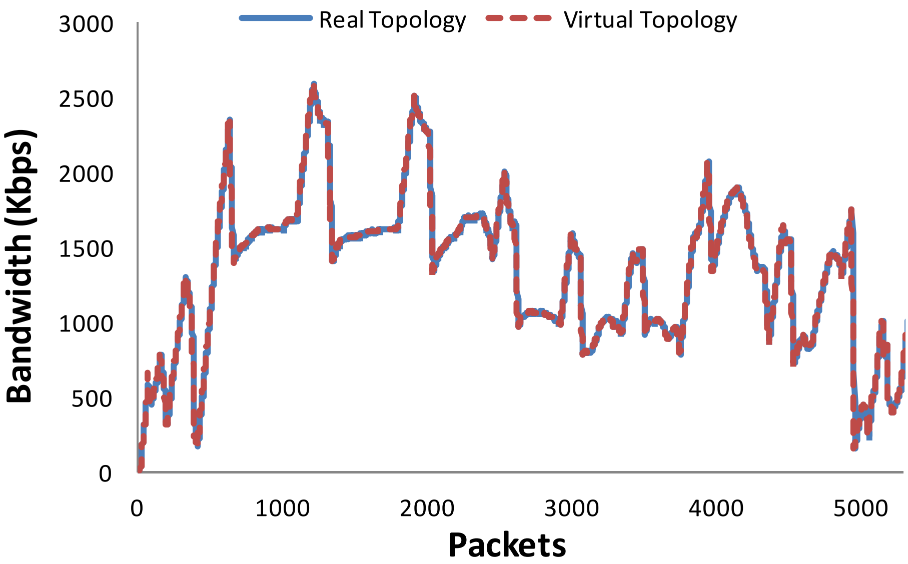


**Figure 5.14.** *Bandwidth in topology 3, WAN link at 8 Mbps.*

### *5.2.6.5 Delay*

We present the results related to the delay obtained for the three topologies.

### *5.2.6.6 Delay in topology 1*

Figure 5.15 shows the delay of the real topology and the delay of the virtual topology. The mean value of the delay in the real topology is 9.34 ms while for virtual topology is 9.32 ms. The minimum values are 0 ms and 0.02 ms for real topology and virtual topology respectively. The maximum values are 94.97 ms and 93.39 ms for real topology and virtual topology respectively. The variance of the data is 75.36 for real topology and 71.36 for virtual topology. The standard deviation is 8.68 in real topology and 8.45 in virtual topology. The percentage of difference between standard deviations is higher than in the bandwidth cases. The data does not follow a normal distribution neither in real or virtual topology.

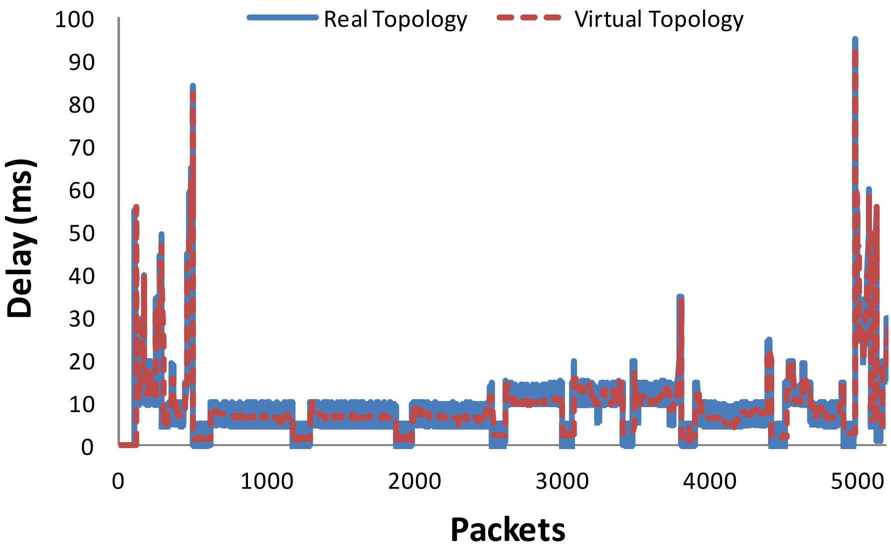


**Figure 5.15.** *Delay in topology 1, link at 1 Gbps.*

### *5.2.6.7 Delay in topology 2*

When we study the delay in topology 2, we can observe two different cases, when the link between switches is established at 10 or 100 Mbps.

In Figure 5.16 we can see the values of the delay of both the real topology and the virtual topology. The data presents similar value for both topologies. The mean value of the delay in the real topology is 9.51 ms while for the virtual topology is 9.39 ms. The minimum and maximum values are 1.11 ms and 94.99 ms for real topology and 1.6 ms and 93.31 ms for virtual topology respectively. The variance of the data is 74.315 for the real topology and 69.427 for the virtual topology. The standard deviation is 8.62 in the real topology and 8.33 in the virtual topology. The data does not follow a normal distribution neither in real or virtual topology.

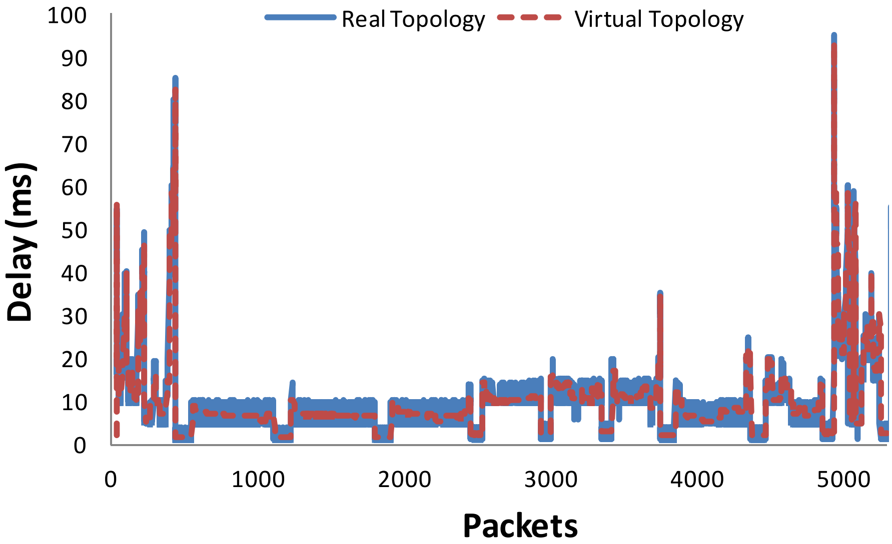


**Figure 5.16.** *Delay in topology 2, link at 10 Mbps.*

Figure 5.17 shows the delay values for both the real topology and the virtual topology. The data presents similar value for both topologies. The mean value of the delay in the real topology is 9.49 ms while for the virtual topology is 9.38 ms. The minimum and maximum values are for the real topology are 0 ms and 95.04 ms, respectively, and 0.96 ms and 93.31 ms for the virtual topology respectively. The variance of the data is 75.24 for real topology and 69.31 for the virtual topology. The standard deviation is 8.67 in the real topology and 8.33 in virtual topology. The data does not follow a normal distribution neither in real or virtual topology.

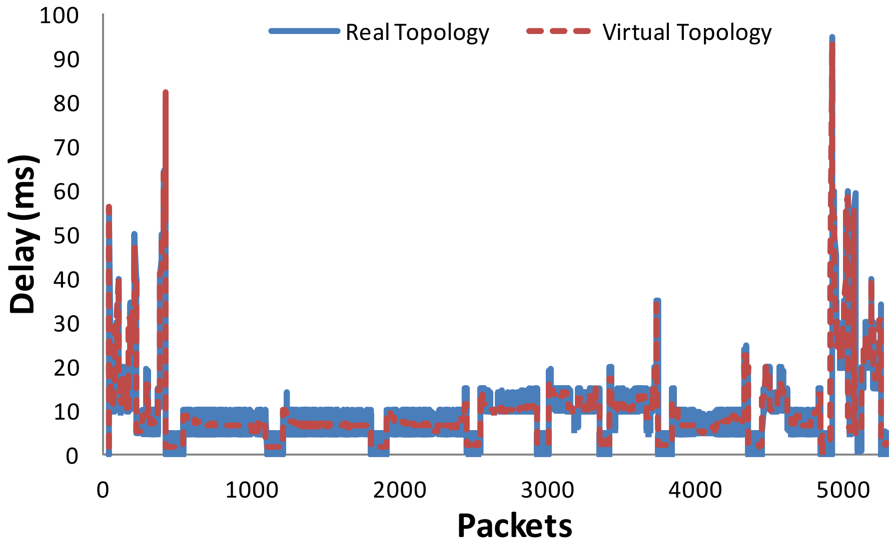


**Figure 5.17.** *Delay in topology 2, link at 100 Mbps.*

#### *5.2.6.8 Delay topology 3*

As in the previous subsection, when we study delay in topology 3, we can observe three different cases, when the WAN link between routers is configured at 2, 4 and 8 Mbps.

In Figure 5.18, it is observed the data gathered about the delay for both topologies, when the WAN link between routers is configured at 2 Mbps. The highest peaks are similar in both cases, but the lowest peaks only appear in the virtual topology. The mean value of delay for the real topology is 9.56 ms, similar to the mean value for the virtual topology, 9.39 ms. The highest delay in real topology is 85.15 ms, and the obtained value for the virtual topology is 82.38 ms. The minimum delay is 5.41ms for the real topology and only 1.69 for the virtual topology. The variance is 63.20 for the real topology and 61.19 for the virtual topology. Regarding to the standard deviation, it is obtained 7.95 for the real topology and 7.82 for the virtual topology. Finally the distribution of both sets of data follows a non-normal distribution.

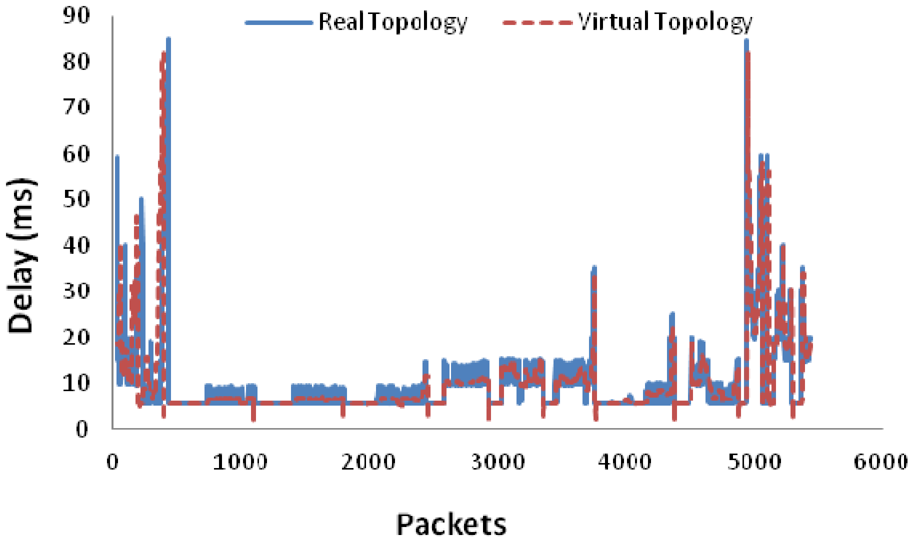


**Figure 5.18.** *Delay in topology 3, WAN link at 2 Mbps.*

The delay of the real and virtual topologies, when the WAN link between routers is configured at 4 Mbps, is shown in Figure 5.19. The peaks are placed in similar position for both topologies, but the real topology presents values in the stable periods. The mean for the real topology is 9.45 ms, quite close to the mean of the virtual topology, which is 9.37ms. The minimum delay is 1.69 ms for the virtual topology and 2.57 ms for the real topology. Regarding to the maximum value, the real topology provides higher value, 95.09 ms (the virtual topology has 93.37ms). The variance is higher for the real topology, 73.45, than for virtual topology, 67.55. The standard deviation is 8.57 for real topology and 8.22 for virtual topology. Finally the data of both topologies follows a non-normal distribution.

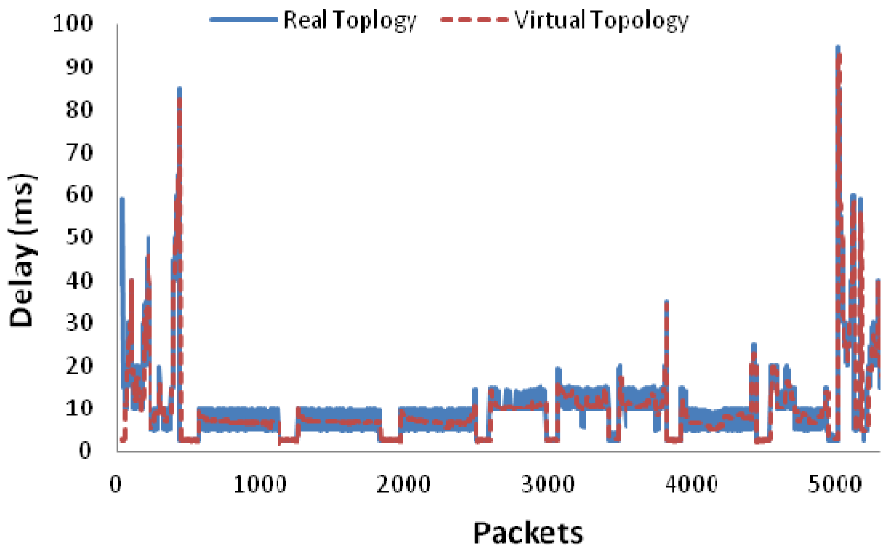


**Figure 5.19.** *Delay in topology 3, WAN link at 4 Mbps.*

In Figure 5.20, it is presented the data about the delay for the virtual and real topologies, when the WAN link between routers is configured at 8 Mbps. The mean value of delay for the real topology is 9.45 ms, similar to the mean value for the virtual topology, 9.41 ms. The highest delay in real simulations is 95.06 ms, which is higher than the delay of the virtual topology (93.43 ms). The minimum delay is 1.20 ms for the real topology, which is lower than the minimum delay for the virtual topology (1.34 ms). Regarding to the variance, it is higher for the real topology, 75.56, than for the virtual topology, 68.46. The standard deviation for the real topology is 8.63 and for the virtual topology is 8.27. Finally the distribution of both sets of data follows a non-normal distribution.

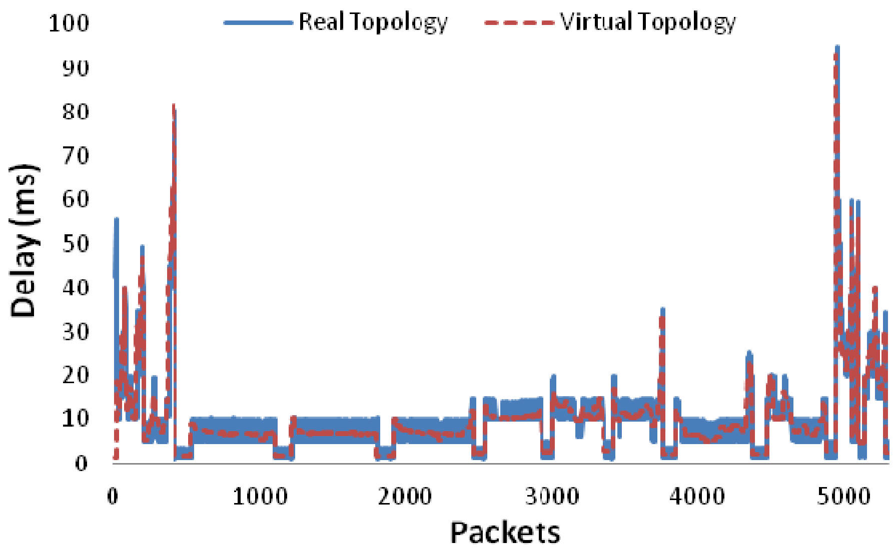


**Figure 5.20.** *Delay in topology 3, WAN link at 8 Mbps.*

#### *5.2.6.9 Jitter*

We present the results related the delay obtained for the three topologies.

#### *5.2.6.10 Jitter in topology 1*

Figure 5.21 shows the values obtained for the jitter in the real topology and in the virtual topology. The values of the real topology are higher than in the virtual topology. The mean value of jitter in the real topology is 0.044 ms, while for virtual topology is lower than 0.004. The minimum and maximum values are the same for both topologies 0 and 3.26. The variance is <0.01 for both topologies. The standard deviation is 0.065 in the real topology and 0.056 in the virtual topology. Respecting to the distribution, the data for both topologies follow a non-normal distribution.

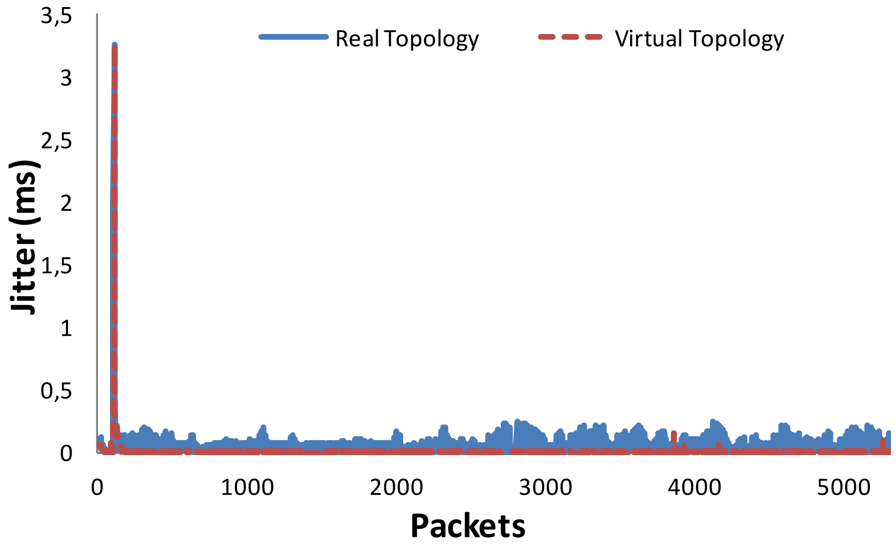


**Figure 5.21.** *Jitter in topology 1, link at 1 Gbps.*

### *5.2.6.11 Jitter in topology 2*

When we study the jitter in topology 2, we can observe two different cases, when the link between switches is set up at 10 or 100 Mbps.

Figure 5.22 shows the values of the jitter in the real topology and in the virtual topology. The values of the real topology are higher than of the virtual topology. The mean value of jitter for the real topology is 0.040 ms, while for the virtual topology is 0.004. The maximum value of the jitter is 1.26 ms for the real topology, while for the virtual topology is 0.84 ms. The minimum value is 0 in both cases. The variance is <0.01 for both topologies. The standard deviation is 0.041 in the real topology and 0.034 in the virtual topology. The data for both topologies follow a non-normal distribution.

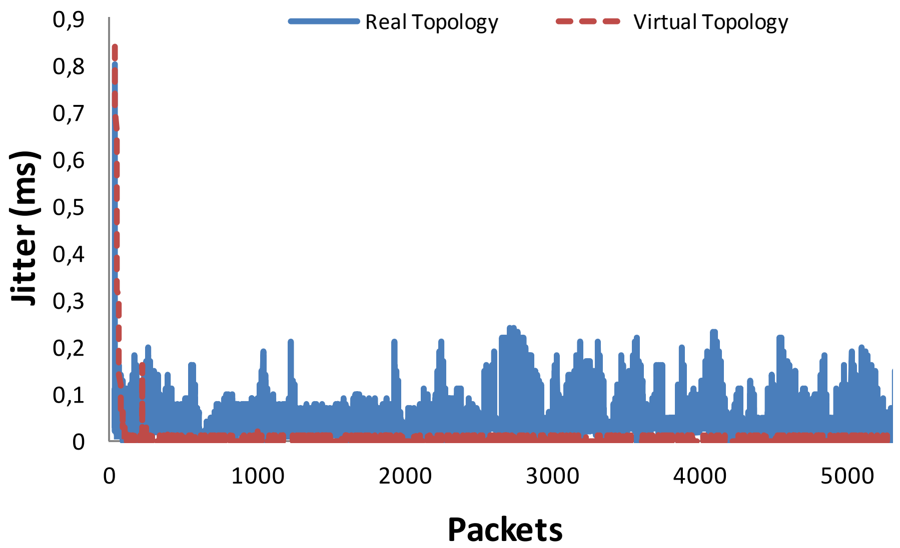


**Figure 5.22.** *Jitter in topology 2, link at 10 Mbps.*

Figure 5.23 shows the values of the jitter in real topology and in the virtual topology. The values of the real topology are higher than of the virtual topology. The mean value of jitter for the real topology is 0.044 ms, while for virtual topology is 0.012 ms. The minimum and maximum values are the same for both topologies 0 ms and 3.33 ms respectively. The variance of the data is 0.0004 for the real topology and 0.0176 for the virtual topology. The standard deviation is 0.07 in the real topology and 0.13 in the virtual topology. The data for both topologies follow a non-normal distribution.

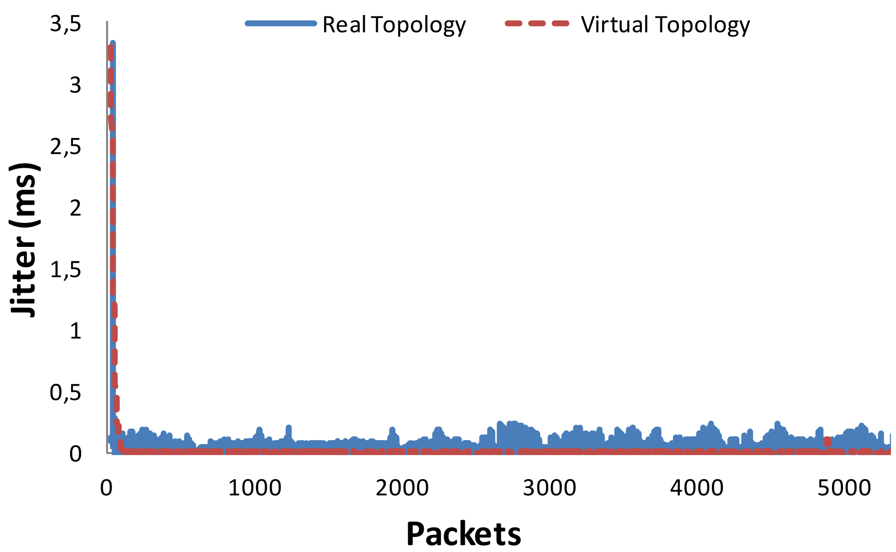


**Figure 5.23.** *Jitter in topology 2, link at 100 Mbps.*

### *5.2.6.12 Jitter in topology 3*

As in previous subsections, when we study the jitter in topology 3, we can observe three different cases, when the WAN link between routers is configured at 2, 4 and 8 Mbps.

In Figure 5.24 we can see the values of the jitter of the real topology and of the virtual topology, when WAN link between routers is configured at 2 Mbps. The data of the real topology presents more peaks than the data of the virtual topology. The mean value of the jitter in real topology is 0.05 ms while the mean value of the virtual topology is 0.03 ms. The minimum and maximum values for the real topology are 0 and 5.25 ms, while for the virtual topology are 0 ms and 5.32 ms. The variance of the data is 0.14 for the real topology and 0.13 for the virtual topology. The standard deviation is 0.14 for the real topology and 0.12 for the virtual topology. The data does not follow a normal distribution neither in real nor virtual topology.

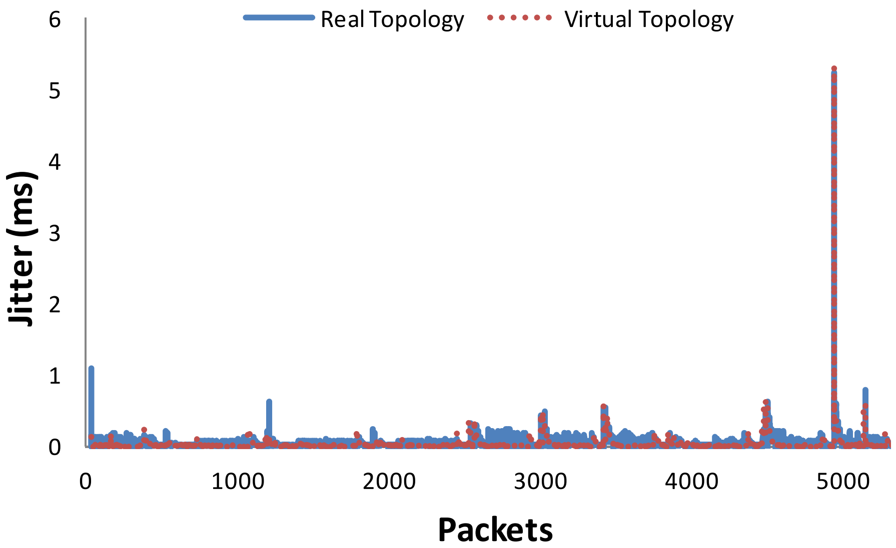


**Figure 5.24.** *Topology 3, WAN link at 2 Mbps.*

We can see the values of the jitter, of the real topology and of the virtual topology, in Figure 5.25, when the WAN link between routers is configured at 4 Mbps. The jitter of the real topology presents higher mean values than in the virtual topology (0.4 ms for the real topology and 0.3 ms for the virtual topology). The minimum and maximum values are 0 ms and 0.67 ms for the real topology and 0 ms and 0.76 ms for virtual topology respectively. Even having higher mean values for the real topology, the maximum value (highest peak) is higher for the virtual topology. Regarding to the standard deviation, the values are almost the same, 0.5 and 0.6 for the real and the virtual topology respectively. The variance of the data is <0.01 in both cases. Finally, for the distribution of the data, we can affirm that the data do not follow a normal distribution.

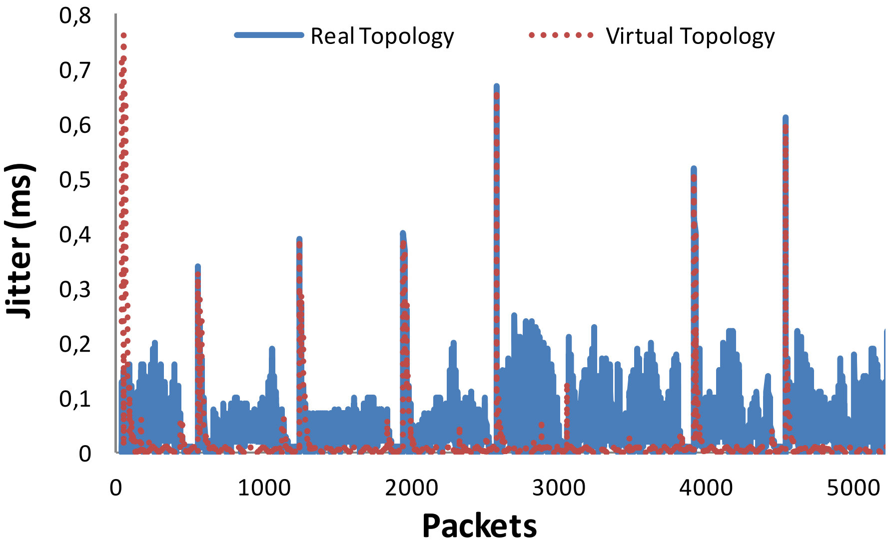


**Figure 5.25.** *Topology 3, WAN link at 4 Mbps.*

Figure 5.26 shows the values of the jitter in real topology and in virtual topology, when the WAN link between routers is configured at 8 Mbps. Values from real topology are higher than in virtual topology. The mean value of the jitter for the real topology is 0.04 ms, while for the virtual topology is lower than 0.01 ms. The maximum value of the jitter is 0.24 ms for the real topology, while for the virtual topology is 0.13 ms, almost half than in real topology. The minimum value is 0 in both cases. The variance is <0.01 for both topologies. As it is expected, the value of the standard deviation is much higher in the real topology, 0.04, than in the virtual topology, which is lower than 0.01. Respect to the distribution, the data for both topologies follow a non-normal distribution.

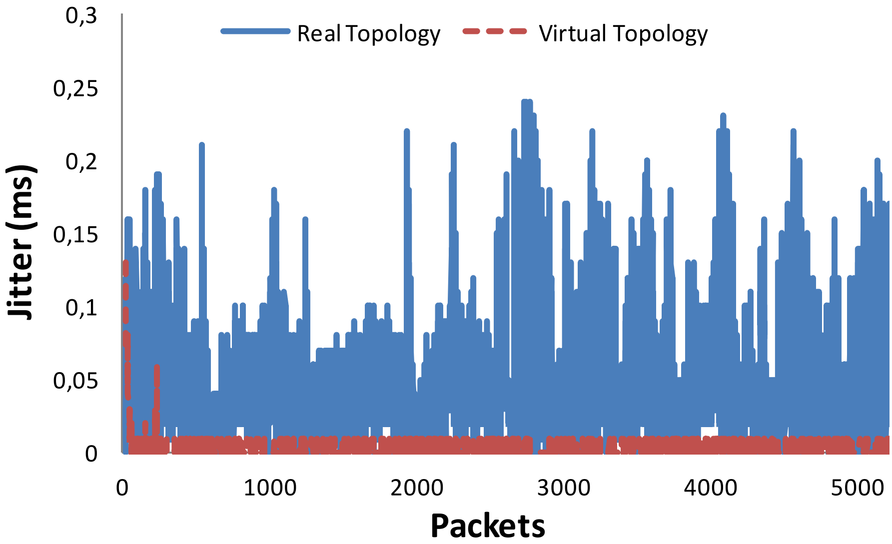


**Figure 5.26.** *Topology 3, WAN link at 8 Mbps.*

## 5.3 QoS and QoE performance test in IPTV video transmission

In order to perform our study we used basic real equipment. Our topology is composed of five routers interconnected by serial links, which simulated WAN connections, five switches that are connected to the FastEthernet interfaces of the routers, which let us create LANs. The topology is shown in Figure 5.27.

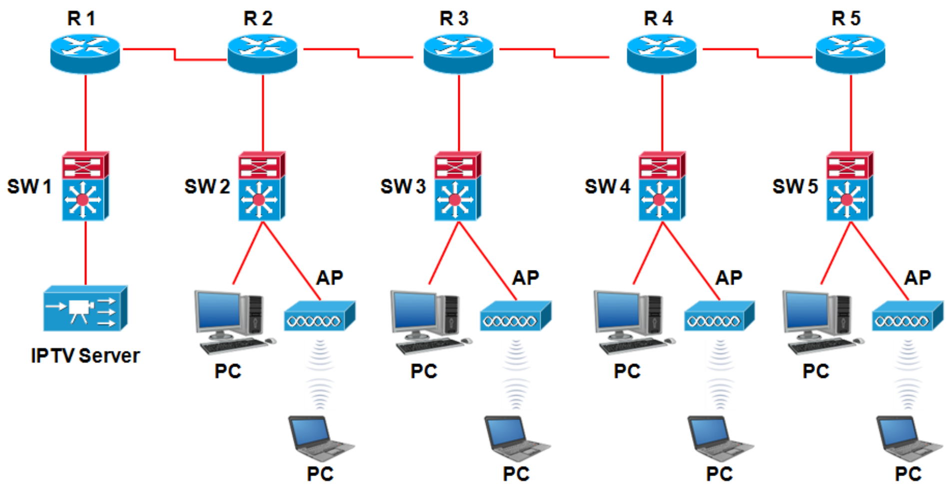


**Figure 5.27.** *Network topology used to perform the test.*

Concretely, the devices used were, routers Cisco 2800 series, two serial WAN and 2 FastEthernet interfaces, multilayer switches Cisco Catalyst 3600 Series, with 24 Fastethernet interfaces and 2 GigabitEthernet interfaces and 2 Cisco Aironet Access Points 1200 Series that support 802.11 a, b and g. To this topology we connected several types of devices such as PCs, laptops, and mobiles at the user side, and at the other side of the topology, we installed an IPTV server, which was used to launch the video transmission.

To stream video and measure the quality at the end user side we used 3 Desktop PCs with the following characteristics. Intel Core i5-2400 CPU @3.10 Ghz with 4 Gb RAM (they had Cisco AIR-PI21AG-E-K9. 802.11 a/b/g Low profile PC adapter and an Intel 82579V Gigabit Ethernet card) and 2 laptops with the following characteristics. Intel Core i7-4510U CPU @3.10 Ghz with 4 Gb RAM. They had an IEEE 802.11 ac+a/g/n Wireless card and a 10/100/1000 Gigabit Ethernet card. Both, desktop PCs and Laptops had Microsoft Windows 7 Professional

In all user devices we installed the VLC media player [173] to receive the video that was sent from the IPTV server, which used VLC as a video streamer. Moreover, we installed other applications that enabled us to obtain more information about received video (Whireshark [174]) and a traffic generator developed by us (Testtool [175]), to observe its consequences. Testtool is able to generate traffic from a computer while the video is being transmitted. In this way we can saturate the bandwidth of the links.

Every time we deliver video for an IPTV channel, the Whireshark application is running in the computer receiving the IPTV channel. It let us capture all packets received. In this way, we obtain relevant information such as the QoS parameters, which are subsequently studied in the evaluation of results phase. Our study has the following phases: codification phase, transmission phase, evaluation of results phase and phase of evaluation of the video quality.

### 5.3.1 Codification phase

The first step we performed in this phase was to select a video for transmission that will be used as a reference video. We took also into account its length to avoid an excessive duration which will make us quicker the phase of evaluation of the received video quality. The video used was Big Buck Bunny, Sunflower version, which is available on the website Bigbuckbunny [176]. We selected 37 seconds where there were several scene changes and movements. We considered this duration enough for our purposes.

To trim and encode the video we used the FFmpeg application. Then, the video was encoded with some of the most used codecs. The selected codecs were Microsoft MPEG-4, XviD, DivX and H.264 / AVC. All codings were performed with a frame size of 1024x768 at different bitrates (the resolution was limited by the maximum transmission capacity of the WAN links of the routers). The video had 60 frames per second and an average variable bit rate of 1496 Kbps.

After making these tests, the codec that showed the best result was H.264 / AVC. We checked it by comparing the sizes of the encoded videos, but taking into account that the final quality are very similar, so the image quality is not going to be compared, because their difference will not be perceived by a regular viewer. As our objective was to work with the smallest file size, which will use the lowest bandwidth during the transmission, we only used videos with H.264/AVC codec.

### 5.3.2 Transmission phase

At this stage we transmit the video from the IPTV server to all customers in the network. Initially, we use the maximum bandwidth allowed by all interfaces. This allows us to take a reference model. Then, we generate five different cases modifying parameters such as bandwidth of the links, delay, bandwidth of the Access Points (AP) and the bandwidth of the wireless clients. In addition, we inject traffic from some PCs, while we transmit the video to the customers in order to see how it affects to the transmission.

At the customers' side, we installed VLC clients to receive the transmitted video, and we also installed Whireshark to capture and analyze the information included in the received video packets.

### 5.3.3 Evaluation of results phase

In this step we study the information obtained through Whireshark. It is gathered in the computers of the customers receiving the IPTV channel. In this case, we really did not observe problems with some network parameters such as delay and jitter. This really happens because the receivers at the customer side use to have a buffer which decreases hugely the impact on these parameters when something happens in the network.

Between all studied cases, only when we decrease the bandwidth on the serial links connecting the routers or when we saturate the available bandwidth, injecting traffic when video is delivered, we observed some problems. Concretely there are packet losses. Table 5.1 shows the packets received, delay, jitter and the percentage of packets lost during transmission, when we vary the bandwidth of the serial interfaces of the routers. All packets are received in all cases except when the available bandwidth is less than the maximum peak value of the bit rate of the video.

**Table 5.1.** Varying the bandwidth between routers

| Clock rate (Mbps.) | Packets (bytes) | Delay (ms) | Jitter (ms.) | Lost Packets (%) |
|---|---|---|---|---|
| **Base line** | 5092 | 7.36 | 1.37 | 0.00 |

| | | | | |
|---|---|---|---|---|
| **2** | 4917 | 7.62 | 2.08 | 3.46 |
| **4** | 5092 | 7.36 | 1.54 | 0.00 |
| **5.3** | 5092 | 7.36 | 1.40 | 0.00 |

Then, we used our tool [175] to generate traffic between two computers, which were neither the IPTV server nor the customer receiving the video. But they were connected to the routers in the path between the IPTV server and the customer. It is shown in Figure 5.28. By injecting traffic, we saturate the bandwidth of the link between routers. Table 5.2 shows the packets received, delay, jitter and the percentage of lost packets during transmission, when it is simultaneously injected traffic in the network through the same path were the IPTV channel is delivered. As in the previous case, there is a packet loss.

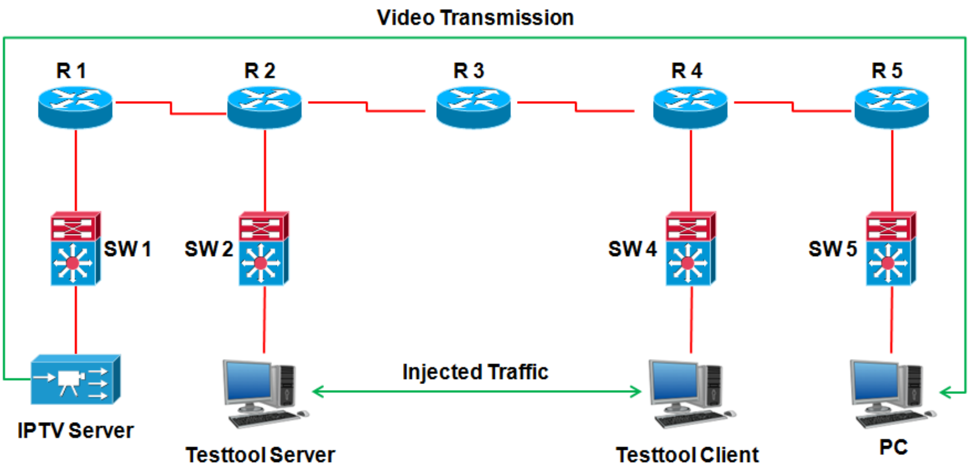


**Figure 5.28.** *Traffic flow between end users through a wired path.*

**Table 5.2.** *Traffic flow in the wired network.*

| Traffic (Mbps.) | Packets (bytes) | Delay (ms) | Jitter (ms.) | Lost Packets (%) |
|---|---|---|---|---|
| **Base line** | 5092 | 7.36 | 1.37 | 0.00 |
| **1** | 5092 | 7.36 | 1.71 | 0.00 |
| **1.5** | 5092 | 7.36 | 1.85 | 0.00 |

| | | | | |
|---|---|---|---|---|
| **2** | 5092 | 7.36 | 2.01 | 0.00 |
| **2.5** | 5090 | 7.36 | 2.00 | 0.04 |
| **3** | 5071 | 7.39 | 2.09 | 0.41 |
| **3.5** | 5034 | 7.44 | 2.15 | 1.14 |
| **4** | 5004 | 7.49 | 2.13 | 1.73 |
| **4.5** | 4955 | 7.56 | 2.05 | 2.69 |
| **5** | 4922 | 7.61 | 1.84 | 3.34 |

Then, we generate traffic using Testtool [175] between two computers connected to the same access point as the customer receiving the video. This case is shown in Figure 5.29. In this case we varied the bandwidth of the AP. Table 5.3 shows the received packets, delay, jitter and the percentage of packets lost during the transmission, when the client was connected using a wireless network, to receive the video, but at the same time there was extra traffic from other wireless customers. As in the previous case, there are packet losses. It also occurs when the available bandwidth is lower than the peak value of the video bit rate.

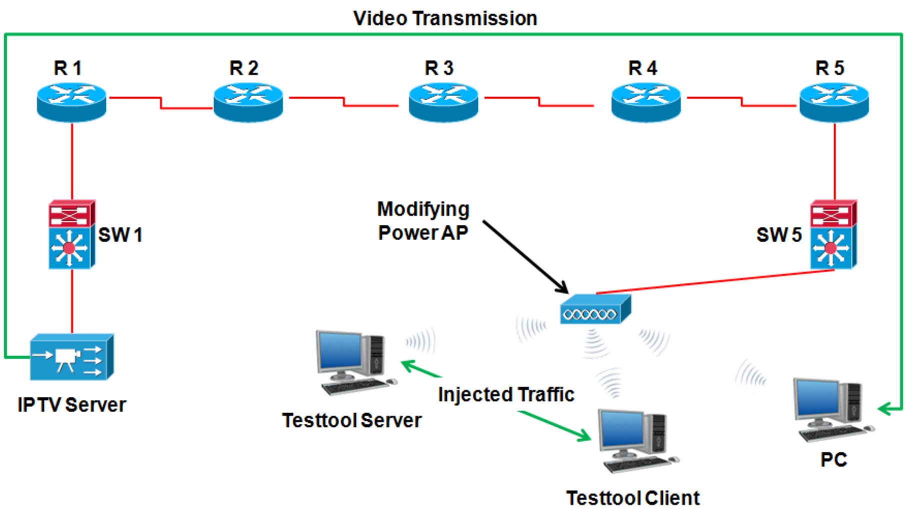


**Figure 5.29.** *Traffic flow between end users when the end user uses a wireless connection.*

**Table 5.3.** *Injecting traffic in the wireless network.*

| Power Trans. APs (mW) | Packets (bytes) | Delay (ms) | Jitter (ms) | Lost Packets (%) |
|---|---|---|---|---|
| **Base line** | 5092 | 7.36 | 1.40 | 0.00 |
| **6** | 5092 | 7.36 | 2.25 | 0.00 |
| **8** | 5085 | 7.37 | 2.57 | 0.14 |
| **10** | 5020 | 7.64 | 4.29 | 1.41 |
| **12** | 4634 | 8.08 | 4.31 | 8.99 |
| **15** | 4153 | 9.02 | 4.91 | 18.44 |
| **20** | 3938 | 9.51 | 5.35 | 22.66 |

### 5.3.4 Phase of evaluation of the video quality

In this phase, we deal the evaluation of the received video at the customer side. We used the DSIS (Double Stimulus Impairment Scale) of the MOS method. We compared the original signal, without degrading, with that obtained at the customer side by a user, which is the degraded signal. The video sequence was viewed twice in each test, and the evaluators waited at the end of each test sequence in order to perform their qualification. The videos are rated from 1 to 5, where 5 is the highest quality value and 1 is the lowest quality value.

We selected a group of evaluators consisting of 20 persons of different sex, age and degree of knowledge of IPTV or video quality. Table 5.4 shows the characteristics of these persons. In all cases in which there had been no packets lost, the score was 5. In the remaining cases, most of them were rated with a score lower than 3, thus the QoE of the end user had an unacceptable level. In almost all cases that were rated with a value below 3, packet loss affected I-type (Intra) images, which are coded without any reference to previous images, so the GOP formed by this I frame and the associated P and B frames, had more degradation.

**Table 5.4.** *Characteristics of the users.*

| Characteristic | | Evaluators |
|---|---|---|
| **Amount** | | **20** |
| **Age** | | |
| | 25 - 29 years old | **4** |
| | 30 - 39 years old | **5** |
| | 40 - 49 years old | **7** |
| | > 50 years old | **4** |
| **Sex** | | |
| | Man | **13** |
| | Woman | **7** |
| **Knowledge about IPTP and Video Quality** | | |
| | Very high | **3** |
| | High | **4** |
| | Medium | **3** |
| | Low | **5** |
| | Very Low | **5** |

One of our tests is shown in Figure 5.30. It shows the values provided by the evaluators when we varied the bandwidth of the serial link between routers. The 20 reviewers rated with a score of 5 the videos received when the available bandwidth between routers was higher than

4 Mbps. But when the bandwidth is set down to 2 Mbps, there were packet loss, so the rating of the quality of video was dropped almost for all reviewers down to 1.

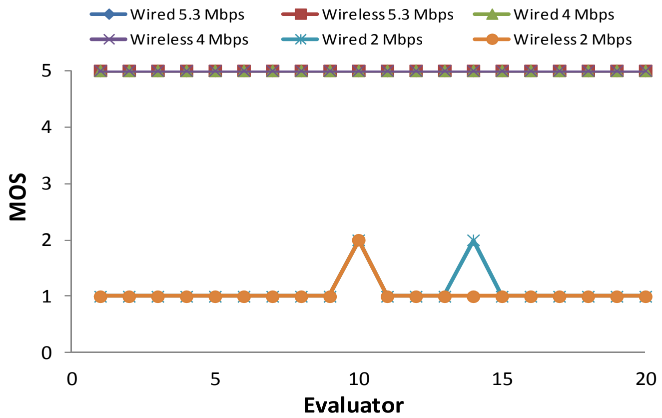


**Figure 5.30.** *MOS assessment when the bandwidth between routers varies.*

Having performed all these tests and after having studied the results, we can ensure that the main problem for an operator providing IPTV service, is the need of enough bandwidth to transmit and receive correctly the information. This will avoid any packet loss.

## 5.4 QoS and QoE performance test in video transmission using logical networks in IPTV

We study the effect of stressing the network. In order to achieve this goal, we study the most well-known network parameters such as latency, jitter and number of lost packets, which provide relevant information to the network operators. We test different cases and analyze the received video quality. It is performed for both wireless and wired networks. First, we get a baseline that is later used to compare different study cases. These cases are designed taking into account the impact of multimedia delivery when there are few network resources.

In order to perform the test bench design, we have taken into account the current technologies used by service providers and the usual features of the networks focused on video streaming. The equipment include a Cisco 2811 Router, a Cisco Catalyst 3560 Switch WS-C3560-24PS-E, a Cisco AIR-AP 1231G-E-K9 Access Point, 3 PC's with Intel Core Processor i5-2400, CPU @3.10 Ghz, 4Gb Memory RAM, Wireless adapter LINKSYS WUSB600N and Ethernet Intel 82579V gigabit network connection and Win7 Operative System, and 1 PC with Intel Core Processor i5-2400 CPU @3.10 Ghz, 4Gb Memory RAM, Wireless adapter Cisco AIR-PI21AG-E-K9, 802.11 a/b/g, Ethernet Intel 82579V gigabit network connection; and Windows XP Professional Service Pack 3 Operative System. The video used to perform the test was the Big Buck Bunny, Sunflower version, which is accessible at [177]. We use FFmpeg program [178] to encode the video with the following features: video time 35 seconds, encoded in H.264 [179], with a resolution of 1024x768 at 30 fps (frames per second) and a bitrate of 1338 Kbps.

We use VLC media player, version 2.1.3, for both server and client. To capture and analyze network parameters, under the Real Time Protocol (RTP), we use Wireshark, version 1.10.5. The main analyzed parameters are the delay (interval between two consecutive packets), jitter (deviation of each packet with respect to the latency), packet loss and Bandwidth.

### 5.4.1 Physical and logical topology

The proposed topology consists of different LANs connected by serial interfaces. The LANs use IEEE 802.3 and IEEE 802.11 technology, while the serial connections use ISO 3309 and ISO 4335 (HDLC) technology. We used Routers Cisco 2811 to connect the LANs. In the LANs we used switches Cisco Catalyst 3560-24PS, which support IEEE 802.1q protocol and Cisco PVST + (Plus per vlan spanning tree). We configured the switch interfaces in different VLANs for each end user. The proposed physical topology can be observed in Figure 5.31.

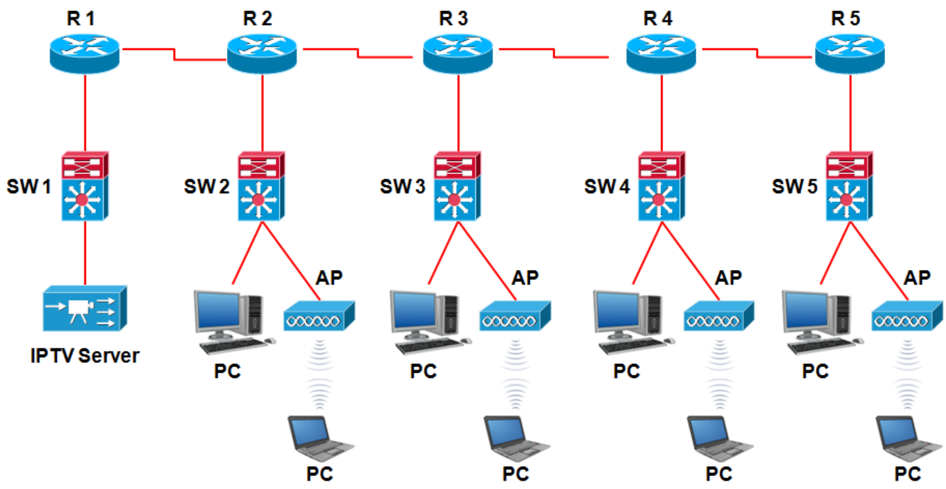


**Figure 5.31.** *Proposed physical topology.*

The logical topology used VLANs, which allow building networks even when they are not sharing the same physical medium. The proposed logical topology can be observed in Figure 5.32.

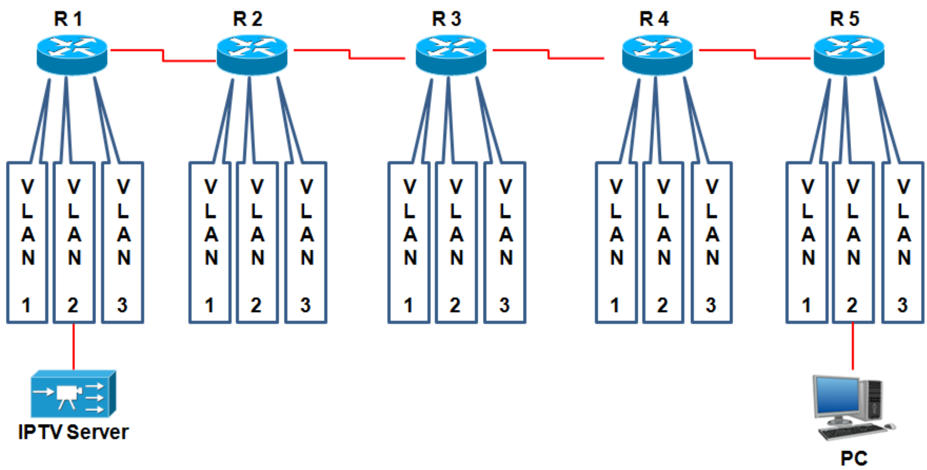


**Figure 5.32.** *Proposed logical topology.*

### 5.4.2 Cases of study and experimentation

In this section we develop a testbed for media streaming, which allow us to perform a series of actions based on network parameters, both internal and external, in order to evaluate the influence on the quality of the received video. The study is divided as follows: first we extract the base line on video streams both wired and wireless, then, we study the effect on the external and external parameters in such streaming.

#### 5.4.2.1 Base Line Study in Wired / Wireless Video Delivery

For the study of the base line we used an IPTV server located at five hops between the VLC Clients (one in the wireless and another in the wired LAN), as shown in Figure 5.33. This study allows evaluating the quality of video in the reception when there is no internal or external factor affecting the transmission delivery. At the reception, we use Wireshark to capture and evaluate the network parameters.

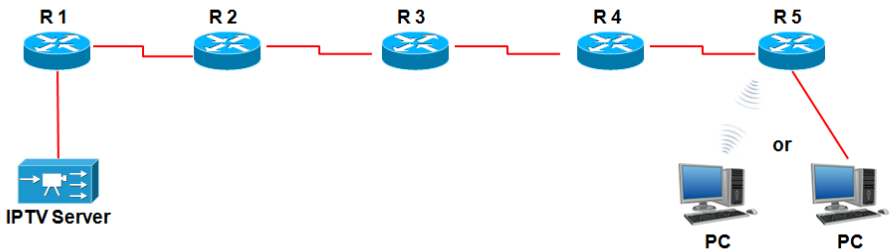


**Figure 5.33.** *Transmission from an IPTV server to a VLC Client (wireless or wired).*

We conduct a series of experiments using this network design to gather measurements of network parameters. They will be the basis for analyzing and comparing them with respect to other study case. For the extraction of these measures we have delivered video through the network and we used Wireshark to capture some network parameters such as jitter, delay, packet loss and bandwidth. These parameters will serve as reference for the analysis of the study case. The results obtained for the delay, in both wired and wireless, are shown in the Figures 5.34 and 5.35. As can it be seen, the delay at the customer side, in both wired and wireless connection, had similar results. It is detected an almost constant delay of approximately 18 ms along the time. Furthermore the highest peak has 130 ms, which is acceptable as it does not exceed the period recommended by the ITU.

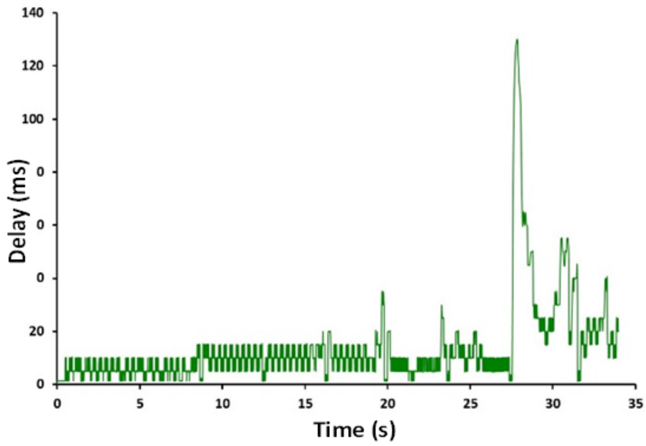


**Figure 5.34.** *Delay of the wired baseline.*

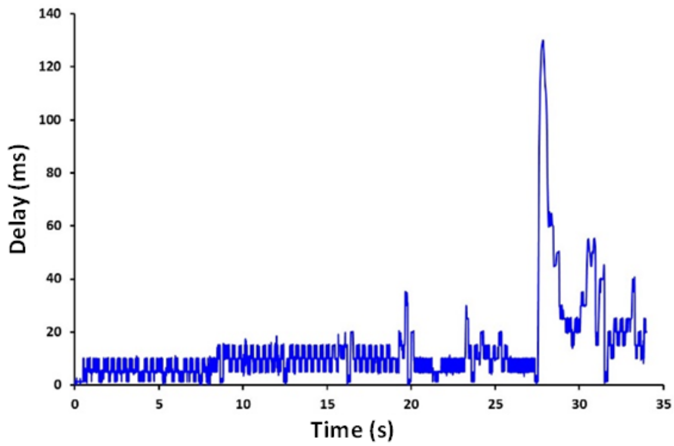


**Figure 5.35.** *Delay of the wireless baseline.*

Figures 5.36 and 5.37 show the jitter at VLC clients side. The results show a peak of 5.5 ms at the beginning of the reception of the video until it stabilizes. Then, the graph follows an almost constant pattern that reaches a value of approximately of 2.5 ms.

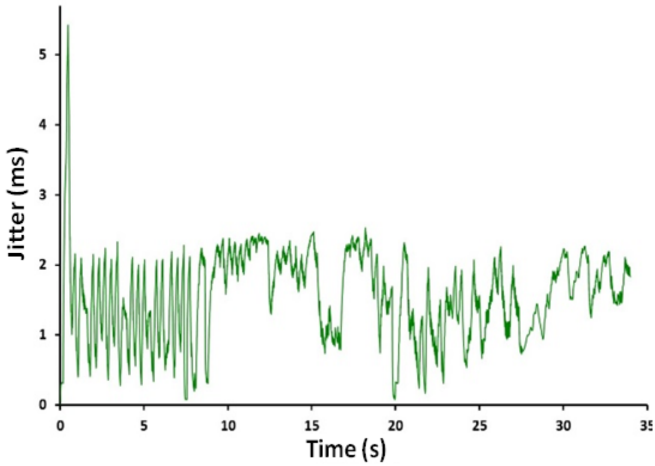


**Figure 5.36.** *Jitter of the wired base line.*

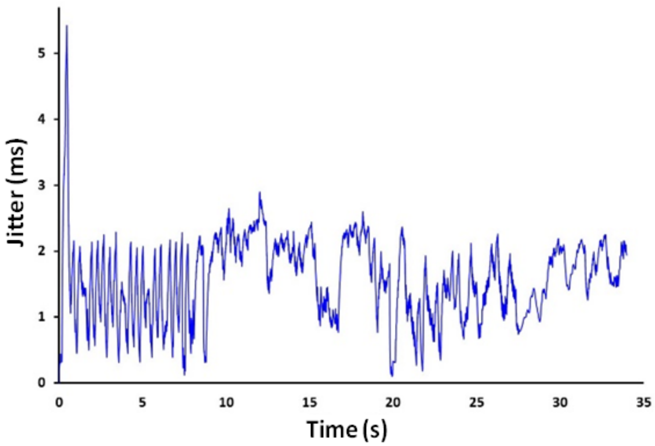


**Figure 5.37.** *Jitter of the wireless base line.*

To estimate the bandwidth over the time, we calculated at each instant the average bandwidth consumed by the last 50 packets received. Considering that the average size of the transmitted packet is 1370 bytes, the bandwidth $BW_t$ at time $T_t$ in which is received the 'i' packet is calculated by the equation (1).

$$BW_i = \frac{N^o\ bits}{Time\ interval} = \frac{1370*50*100000}{(T_i - T_{i-50})*1000} \quad (1)$$

Where, Ti-50 indicates the time instant reception of packet number 'i-50'. Factors 100000 and 1000 have been included to normalize the time, which originally are given in tens of microseconds, and bits units respectively, so that the result is displayed in Kbps. Figures 5.38 and 5.39 show the estimated bandwidth timely manner at each instant of time. It is possible to observe that in both cases, the mean values consumed at the reception, for both wired and wireless network, range approximately 2 Mbps. There are also values that can exceed 9 Mbps. Finally, remark with respect to the loss of packets, that all transmitted packets (5092 packets), have been successfully received. This data will also be included in the baseline. Since neither losses of packages nor mistakes exist in the transmission, we obtained an optimal quality of video reception.

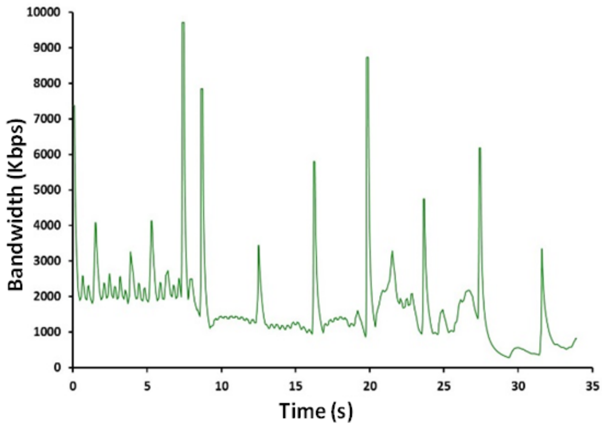


**Figure 5.38.** *Bandwidth wired base line.*

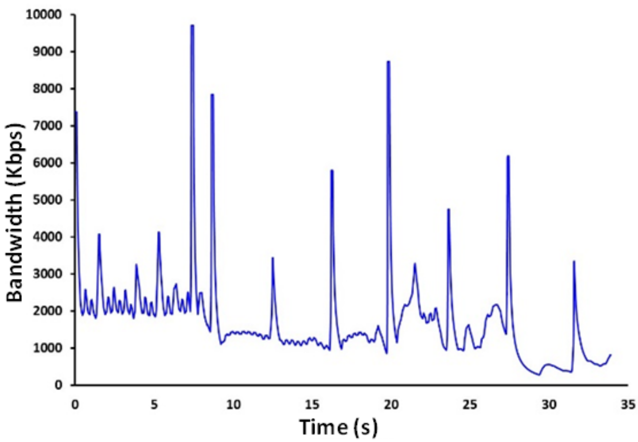

**Figure 5.39.** *Bandwidth wireless base line.*

#### *5.4.2.2 Study of the Internal Network Parameters and the Transmission Effects*

We study the effect on the multimedia traffic. Concretely, we study, the video quality perceived when we modify internal parameters of the network configuration and protocols such as the transmission rate of WAN links, and when we decrease the available bandwidth by increasing other traffic type.

*The Effects of Transmission Speed in WAN Links - Point to Point are the following.*

In this first case study we measure the video reception, in both wired and wireless environments, by varying the transmission speed of the WAN links. The values of the bandwidth selected for the study are 2 Mbps, 4 Mbps and 5.3 Mbps. The design proposed for the studio is observed in Figure 5.40.

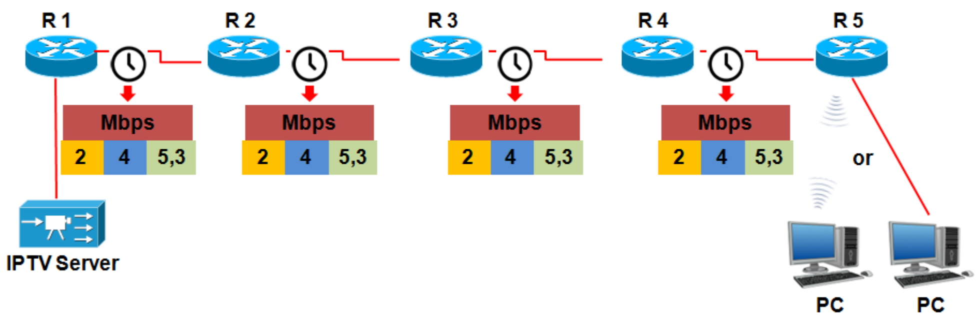


**Figure 5.40.** *Transmission from the IPTV server to the VLC Client using different WAN links.*

After the analysis of the network parameters captured during video delivery through the WAN links at different speeds, we obtained the following results. Regarding the delay, we observe that at a transmission speed of 2 Mbps in the WAN links, the average value obtained is 6.2 ms. However, for 4 Mbps and 5.3 Mbps the average was 7.36 ms, in both wired and wireless (it is shown in Figure 5.41).

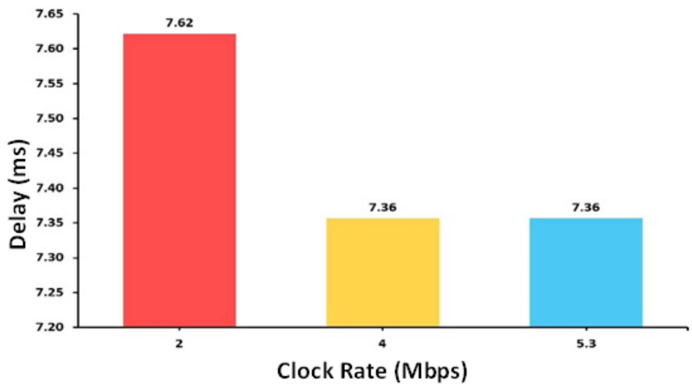


**Figure 5.41.** *Average delay at different transmission speeds in the WAN links.*

Concerning the jitter, we obtain different values for wired than wireless. For a transmission rate of 2 Mbps in WAN links, the obtained average results are superior when the links are 5.3 Mbps and 4 Mbps, as it can be observed in Figure 5.42.

To determine the average percentage of lost packets, we performed the following calculations. First, we subtracted the sequence of packets between the last and the first visible package. Then, the number of packets is obtained, which in this case is 5092. Later, we estimated the difference between the sequence number of packets minus the number of packets (this is the number of lost packets). Finally, we divide the number of lost packets by the packet sequence, thus obtaining the percentage of lost packets.

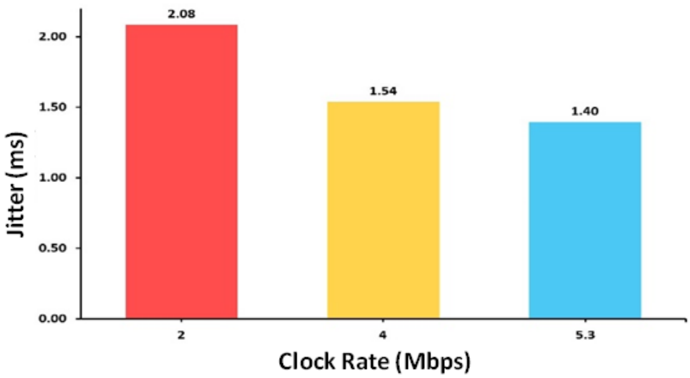


**Figure 5.42.** *Average jitter at different transmission speeds in the WAN links.*

We can see in Figure 5.43 that with a transmission rate of 2 Mbps at WAN links, the average percentage of lost packets is 3.44% for wired environments, and 3.48% for wireless. For a transmission speed of 4 Mbps and 5.3 Mbps in WAN links, we obtained an overall average of 0%. This means that there are no packet losses for both cases in the reception. Given that there is a percentage of loss in estimating the transmission speed of 2 Mbps WAN link, we evaluate in this particular case which is the number of packets lost during the delivery and thus make a proper balance.

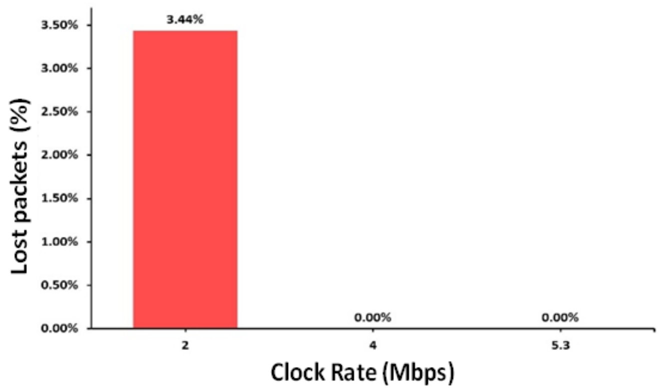


**Figure 5.43.** Average lost packets for different transmission speeds of the WAN links.

In Figure 5.44, it is shown the average bandwidth of the video streams obtained when we vary the transmission speed in WAN links at 2 Mbps, 4 Mbps and 5.3 Mbps. For a WAN link with 2 Mbps, there was an average value of 2181 Kbps. For 4 Mbps and 5.3 Mbps we observed an average value of 2456 Kbps.

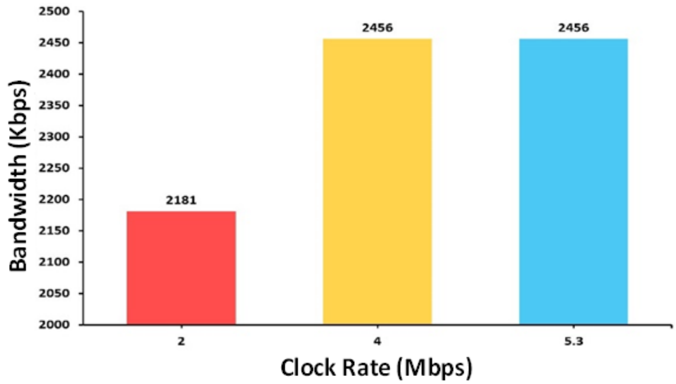


***Figure 5.44.*** *Average bandwidth for different transmission speeds of the WAN links.*

### 5.4.3 Result Analysis

In this section, we analyze the results we have obtained in the previous section. We carry out an analysis of the statistical variance to compare the effect of the studied parameters in relation to the base line in order to test the null hypothesis H0 defined in each case. The null hypothesis H0 is defined as follows. H0 is met when there is no statistically significant difference among the parameters: average latency, jitter, packet loss or bandwidth of each measure compared to those in the baseline. A value of $\alpha = 0.01$ is established to carry out the calculation. We focus our study only on the analysis of the internal parameters in multimedia transmissions, because the results of the effect of external ones have not been conclusive. We compare the results obtained in these cases with the baseline.

In this case, we change the transmission speed of the WAN links in the wired LAN and study the QoS parameters. The statistical calculations summary of the delay is shown in Table 5.5. If we establish the transmission speed of the WAN links to 2Mbps the average delay value is 7.62 ms. Statistically, it shows a significant difference, but from the point of view of QoS, it may be considered negligible or insignificant. Likewise, we cannot exclude the results with the other experimental conditions: 4 Mbps and 5.3 Mbps. Therefore, it indicates that with those values an increment of the link bandwidth does not affect the latency behavior respect to the one obtained in the baseline.

**Table 5.5.** *Study of the delay when comparing the baseline with different transmission speeds of the WAN Links.*

| ransmission speed of WAN links (Mbps.) | Network parameters | | Statistical parameters | | |
|---|---|---|---|---|---|
| | umber of packets (bytes) | elay (ms) | andard Deviation (ms) | Subhead | |
| | | | | in. (ms) | ax. (ms) |
| **Base Line** | 5092 | 7.36 | 7.10 | 7.10 | 7.61 |
| **2** | 4917 | 7.62 | 5.76 | 7.41 | 7.83 |
| **4** | 5092 | 7.36 | 6.86 | 7.11 | 7.60 |
| **5.3** | 5092 | 7.36 | 7.03 | 7.10 | 7.61 |

In Table 5.6, we show the summary of the statistical calculations for the jitter. The average jitter value is 2.08 ms. when we establish the transmission speed of the WAN link to 2 Mbps and 1.54 ms when we establish it to 4 Mbps. The jitter obtained for 2 Mbps and 4 Mbps is out of the confidence interval range established in the baseline measure. Therefore, we found in this experiment, with a 99.9% of guarantee, that when the transmission rate of the WAN links is 2 Mbps or 4 Mbps, there is a significant increase of the jitter. Likewise, the jitter when there is a transmission speed of 5.3 Mbps in the WAN links is 1.40 ms. There is a significant difference from the statistical point of view. However, we consider it negligible from the viewpoint of the QoS.

**Table 5.6.** Study of the jitter when comparing the baseline with different transmission speeds of the WAN Links.

| ransmission speed of WAN links (Mbps.) | Network parameters | | Statistical parameters | | |
|---|---|---|---|---|---|
| | umber of packets (bytes) | itter (ms) | andard Deviation (ms) | Subhead | |
| | | | | in. (ms) | x.(ms) |
| **Base Line** | 5092 | 1.37 | 0.70 | 1.34 | 1.39 |

| ransmission speed of WAN links (Mbps.) | Network parameters | | Statistical parameters | | |
|---|---|---|---|---|---|
| | | | | Subhead | |
| | umber of packets (bytes) | itter (ms) | andard Deviation (ms) | in. (ms) | x.(ms) |
| **2** | 4917 | 2.08 | 1.55 | 2.03 | 2.14 |
| **4** | 5092 | 1.54 | 0.83 | 1.51 | 1.57 |
| **5.3** | 5092 | 1.40 | 0.75 | 1.37 | 1.42 |

The summary of the statistical calculations for packet loss in shown in Table 5.7. The average value of packet loss is 3.46% when the transmission speed of the WAN links is established to 2 Mbps. That value is out of the confidence interval range established in the baseline measure. Therefore, we found in this experimental condition, with a 99.9% of guarantee, that when the transmission speed of the WAN links is 2 Mbps, there is a significant increase in the number of packet loss. We cannot exclude the difference of the other experiments: 4 Mbps and 5.3 Mbps. They indicate that an increment of the bandwidth in the WAN link, respect to those initially shown in the baseline, does not affect to the packet loss. To sum up, in this first case, we can see that the transmission is affected when the transmission speed of the WAN links is reduced to 2 Mbps. The most affected parameter is the packet loss. Jitter is affected in a lesser grade. Delay is not affected.

**Table 5.7.** *Study of the packet loss when comparing the baseline with different transmission speeds of the WAN Links.*

| Transmission speed of WAN links (Mbps.) | Network parameters | | Statistical parameters | | |
|---|---|---|---|---|---|
| | | | | Subhead | |
| | Number of packets (bytes) | Packet loss (ms) | Standard Deviation (ms) | Min. (ms) | Max. (ms) |
| **Base Line** | 5092 | 0.00 | 0.00 | 0.00 | 0.00 |
| **2** | 4917 | 3.46 | 18.27 | 2.79 | 4.13 |

| Transmission speed of WAN links (Mbps.) | Network parameters | | Statistical parameters | | |
|---|---|---|---|---|---|
| | Number of packets (bytes) | Packet loss (ms) | Standard Deviation (ms) | Subhead | |
| | | | | Min. (ms) | Max. (ms) |
| **4** | 5092 | 0.00 | 0.00 | 0.00 | 0.00 |
| **5.3** | 5092 | 0.00 | 0.00 | 0.00 | 0.00 |

## 5.5 Performance test of IP Video Delivery for mobile devices

In order to carry out these tests, we implement a simple topology composed by a video server, a switch that acts only as a transparent gateway and a wireless access point. The wireless network uses the IEEE 802.11 g variant. Transmissions are made in a controlled laboratory environment where the only traffic captured is the one that intervenes in the connection of the devices, as well as in the video transmission. To avoid the generation of DHCP and ARP traffic, static addressing is used to identify our devices in the network

The server uses the VLC media player software to launch the videos. The clients need to use VLC Media Player to locally reproduce the videos and a network sniffer that controls the traffic and picks it up in * .pcap files. The mobile devices have used the PacketCapture application for android while the laptops have used Wireshark.

Figure 5.45 shows the topology used and the address used in each device.

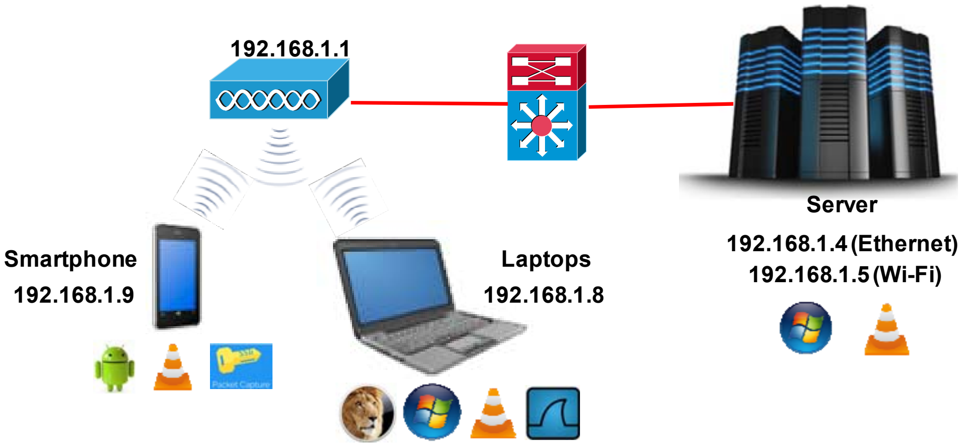


**Figure 5.45.** *Topology used in our test bench.*

### 5.5.1 Devices and video characteristics

This section presents the different devices that have been used to develop these tests as well as their hardware and software features. The characteristics of the videos used are also presented.

As Figure 5.45 shows, to perform our tests, we used a desktop computer that acts as a server and portable devices that will act as the client that sends the video request to the server. As Table 5.8 shows, 4 laptops with different hardware features are used. All of them, with the exception of the Apple MacBook, run Windows 7 as the operating system (OS).

**Table 5.8.** *Characteristics of the laptops and server used in this test bench.*

| Devices | Server | Apple MacBook | Asus Eeee PC Seashell series | ny Vaio PCG-51312M | HP Pavilion dv6 |
|---|---|---|---|---|---|
| **CPU** | QuadCore 3,3 Ghz | Intel Core 2 Duo 2,4 Ghz | Intel Atom N450 Processor *1 1,66 Ghz | al Core 1,2 Ghz | AMD Phenom QuadCore 2 GHz |
| **RAM** | 4 GB 1333 DDR3 | 4 GB 1333 Mhz DDR 3 | 1 GB DDR2 DIMM | GB 1333 Mhz DDR3 | 2 GB 1333 DDR3 SDRAM |
| **O S** | Windows 7 Professio nal 64 bits | OS X 10.7.5 "Lion" 64 bits | Windows 7 Starter 32 bits | Windows 7 Ultimate SP1 64 bits | Windows 7 Ultimate SP1 64 bits |
| **Graphic** | Intel HD Graphics | GeForce 9400M NVIDIA | Atheros 802.11n Wireless | ATI Mobility Radeon HD | ATI Mobility Radeon |

| | | | | | |
|---|---|---|---|---|---|
| | Family | | | 4500 | HD 5650 |
| **Wireless Card** | Linksys WUSB600N Wireless n | Atheros AR9285 802.11b/g WiFi Adapter | WLAN 802.11 a/b/g/n@2.4 GHz | heros AR9285 | IEEE 802.11b/g/n |
| **Screen inches** | 21,5" | 13" | 10,1" | 13,3" | 15,6" |
| **Resolution** | 2560x1600 | 1280x800 | 1024x600 | 1366x768 | 1376x768 |

Additionally, in this bench test, we also want to check the behavior of a mobile device such as a mid-range smartphone. Table 5.9 shows the characteristics of the Smartphone used. To perform the test we used a Google Nexus 5 that runs Android as OS.

**Table 5.9.** *Characteristics of the smartphone used in this test bench.*

| Device | CPU | RAM | O S | Graphic Card | Wireless Card | Screen inches | Resolution |
|---|---|---|---|---|---|---|---|
| **Google Nexus 5** | QuadCore 2,3 Ghz | 2 GB | Android OS, v 5.0 | True HD IPS+ | Wifi 802.11 | 4,95" | 1920x1080 |

To perform the test, 2 videos with a duration of 50 seconds have been used. Table 5.10 shows the characteristics of both videos.

**Table 5.10.** *Characteristics of the used videos.*

| PARAMETER | Video 1 | Video 2 |
|---|---|---|
| **Resolution** | 1920x1080p | 1024x768p |
| **Frame Rate** | 60 fps | 30 fps |
| **Size** | 21,4 MB | 7,97 MB |
| **Time** | 50s | 50s |
| **Bitrate** | 3585 Kbps | 1337 Kbps |
| **Format- Codec** | AVC MPEG-4 | AVC MPEG-4 |
| **Scale** | 16:9 | 16:9 |
| **Chroma** | 4:2:0 | 4:2:0 |
| **Space Color** | YUV 8 bits | YUV 8 bits |

### 5.5.2 Results

This section shows the results obtained after sending both videos and being reproduced with the devices presented in the previous section.

#### 5.5.2.1 *Results for laptops*

Figure 5.46 shows the measured bandwidth for the ASUS Atom device for both videos. As we observed, the bandwidth registered when video 1 is sent reaches maximum values of 300 Kbps, registering a peak higher than 400 Kbps at 20th second. From this point, the band width remains around 200 Kbps. On the other hand, the bandwidth registered when video 2 is sent reaches maximum values of 1100 Kbps.

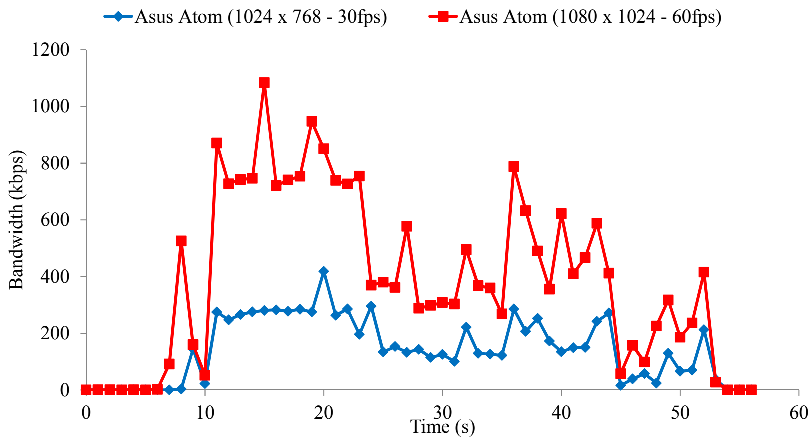


**Figure 5.46.** *Bandwidth for the ASUS Atom device for both videos.*

Figure 5.47 shows the measured bandwidth for the HP Pavilion laptop when both videos are reproduced. In this case, the bandwidth recorded when video 1 is sent reaches maximum values of 300 Kbps, registering two peaks that reach 400 Kbps in the second 12 and 20, respectively. From 25th second, the bandwidth remains around 180-190 Kbps. On the other hand, the bandwidth registered when video 2 is sent reaches maximum values of 1050kbps with an average value between the second 10 and 25 of 800 Kbps. The rest of the bandwidth values oscillate between 400 Kbps and 750 Kbps.

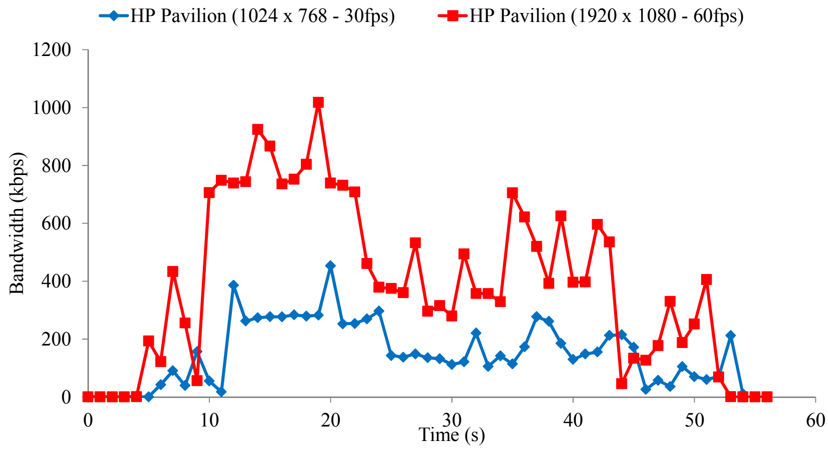


**Figure 5.47.** *Bandwidth for the HP Pavilion device for both videos.*

Figure 5.48 shows the measured bandwidth for the MacBook laptop when both videos are transmitted. The bandwidth recorded when the video 1 is sent reaches average values of 300 Kbps between the second 12 and 20. From this moment, the bandwidth remains around 180-190 Kbps. On the other hand, the bandwidth registered when video 2 is sent reaches maximum values (3 peaks are observed) of 1100 Kbps. The rest of the bandwidth values oscillate between 400 Kbps and 750 Kbps.

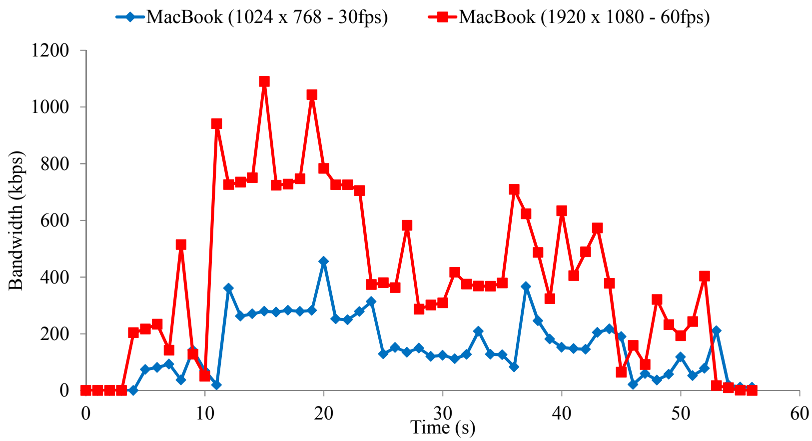


**Figure 5.48.** *Bandwidth for the MacBook device for both videos.*

Figure 5.49 shows the measured bandwidth for the Sony Vaio laptop when both videos are being transmitted. The bandwidth recorded when the video 1 is sent oscillates between 200 Kbps and 300 Kbps up to the second 35. After that, the bandwidth oscillates between 100 Kbps and 300 Kbps. Regarding to the transmission of video 2, the bandwidth presents the same behavior as the rest of laptops.

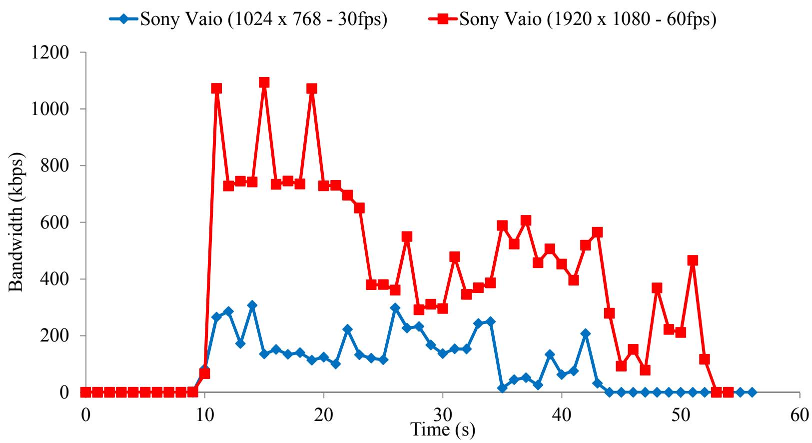


**Figure 5.49.** *Bandwidth for the Sony Vaio device for both videos.*

### 5.5.2.2 Results for mobile devices

Figure 5.50 shows the measured bandwidth when we use the Google Nexus 5 Smartphone to play both videos. In both cases, the behavior is very similar, registering 2 peaks close to 2 Kbps at the beginning of transmission. From the second 10, several oscillations are observed between few bytes to 0.15 bytes for the case of video 1 and from 0 to 0.5 bytes for the case of video 2.

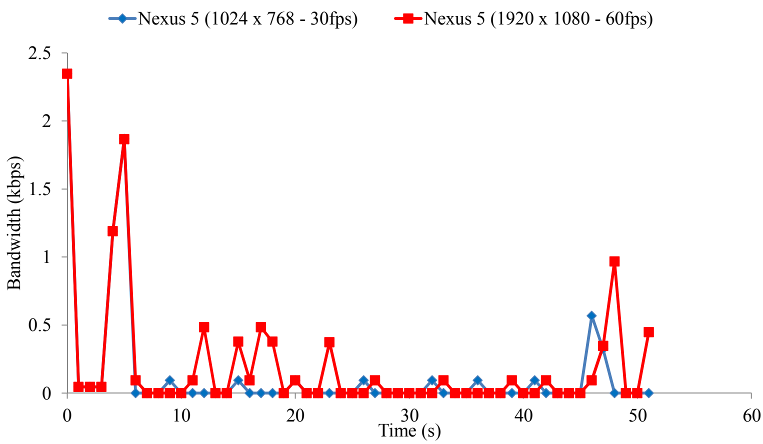

**Figure 5.50.** *Bandwidth for the Google Nexus 5 mobile device for both videos.*

### 5.5.3 Discussion

In order to draw some conclusion about the performance as a function of the hardware features, it is important to compare the device transmissions for the same video.

As we can see in Figures 5.46 to 5.49, the bandwidth of all devices when video 1 is transmitted is very similar for 3 of the 4 devices. The device that presents the worst performance is Sony Vaio device. On the one hand, it seems to be a little less stable than the rest of the devices. In addition, in the rest of the cases, we can observe a flat area of approximately 10 seconds during which the request and beginning of the transmission of the video is made. This flat area reaches approximately 250 Kbps. After this interval of time, the bandwidth stabilizes around 200 Kbps, registering some peaks that can reach 250 Kbps. In the case of the Sony Vaio, this flat area starts with an extension of approximately 25 seconds with oscillating values between 180 and 220 Kbps. The rest of the transmission oscillates between the few kbps and the 200 Kbps.

In the case of the transmission of video 2, the results are almost identical. In all cases, there is an area of about 15 seconds with values of 780 Kbps where 3 peaks that reach 1200 Kbps are registered. The rest of the transmission is maintained around 450-500 Kbps.

Finally, it should be noted that the Sony Vaio laptop is the device that presents worse performance for the transmission of video 1 despite not being the worst hardware features. The Sony Vaio laptop presents a bandwidth much lower than the rest of devices. This fact can also be observed in table 5.11, where the average bandwidth values are shown. As we observed, all the devices are kept in similar values, in exception of the Sony Vaio device.

On the other hand, we observe that despite of making the video requests for all cases in the same way, the traffic recorded by the Smartphone is very similar, despite increasing the resolution of the video.

**Table 5.11.** *Summary of the average values of bandwidth obtained in Kbps.*

| Kind of Device | Video Resolution and Frame Rate | Device Model | Average Bandwidth (Kbps) |
|---|---|---|---|
| **Laptops** | 1024 x 768 - 30fps | Asus Atom | 142.2 |
| | | HP Pavilion | 148.81 |
| | | MacBook | 137.59 |
| | | Sony Vaio | 106.46 |
| | 1920 x 1080 - 60fps | Asus Atom | 394.56 |
| | | HP Pavilion | 391.81 |
| | | MacBook | 406.53 |
| | | Sony Vaio | 386.92 |
| **Mobile Devices** | 1024 x 768 - 30fps | Nexus 5 | 0.14 |
| | 1920 x 1080 - 60fps | Nexus 5 | 0.19 |

## 5.6 Performance test of IP Video Delivery for Heterogeneous Networks using HTML5

In order to provide interoperability, all browsers should be compatible with the same codecs. The problem is that no one meets the requirements requested by the browsers' developers (freedom of licenses per unit or distributor, compatible with the open source model, quality for use, and free of additional risk with the patents of the most important companies).

### 5.6.1 Description of the test bench

In order to start making measurements we took a raw video without any type of compression. Then, we compress it with different codecs and bitrates and we measured the size of the compressed files and the compression time. These data let us analyze and draw conclusions with regard to the used codecs.

The sequence of images has been taken from a Blackmagic Cinema camera that has the following specifications: shooting Resolution: ProRes and DNxHD at 1920x1080, active sensor size 15.81mm x 8.88mm, dynamic range 13 stops, storage type removable 2.5" SSD, SDI Video Outputs 1x10 bit HD-SDI 4:2:2 with choice of film or video Dynamic Range. When recording is set to 25p or 29.97p with SDI Overlays switched off, SDI output format is 1080i50 and 1080i59.94 respectively.

The computer used for data processing, the measurements, and coding was a HP Pavilion dv6 Notebook PC with QuadCore AMD Phenom, 2000 MHz processor. ACPI Architecture X64-based, PC (Mobile) AMD M785, AMD K10 with 2GB DDR3-1333 DDR3 SDRAM and cache memory L1 (512 Kb) L2 (2048 Kb). The Operating System used was Microsoft Windows ® 7 SP2 with DirectX 11.0. In addition it has an ATI Mobility Radeon HD 5650 1024MB graphic card, AHCI 1.0 Serial ATA, network Atheros AR9285 802.11b/g Wifi Adapter and Realtek RTL 8168D/8111D PCI-E Gigabit Ethernet. The software used for the encoding of files was the VLC media player 2.1.3 RinceWind. We discarded Sorenson Squeeze Premium 9.0.2.81 since it still does not provide support for Ogg coding. The video used to perform our test had the next features:

- Video file Size: 597 MB
- Duration: 6s 298ms
- Bitrate: 795 Mbps
- Color space: YUV
- Frame Size: 1920 pixels (width) x 1080p pixels (height)
- Aspect: 16:9
- Frame rate: NTSC 29,976
- Chroma subsampling: 4:2:2

- Compression loss: less mode.

In Figure 5.51 is shown the process of requesting a multimedia resource by several users. All users will perform a http request to the server in order to see the video content and the server will provide the appropriate video according to the user's browser. HTML5 implements bookstores (x264, libtheora and ffvpx) through the browser by facilitating to play the video sent by the server. We will detail the system in the next section.

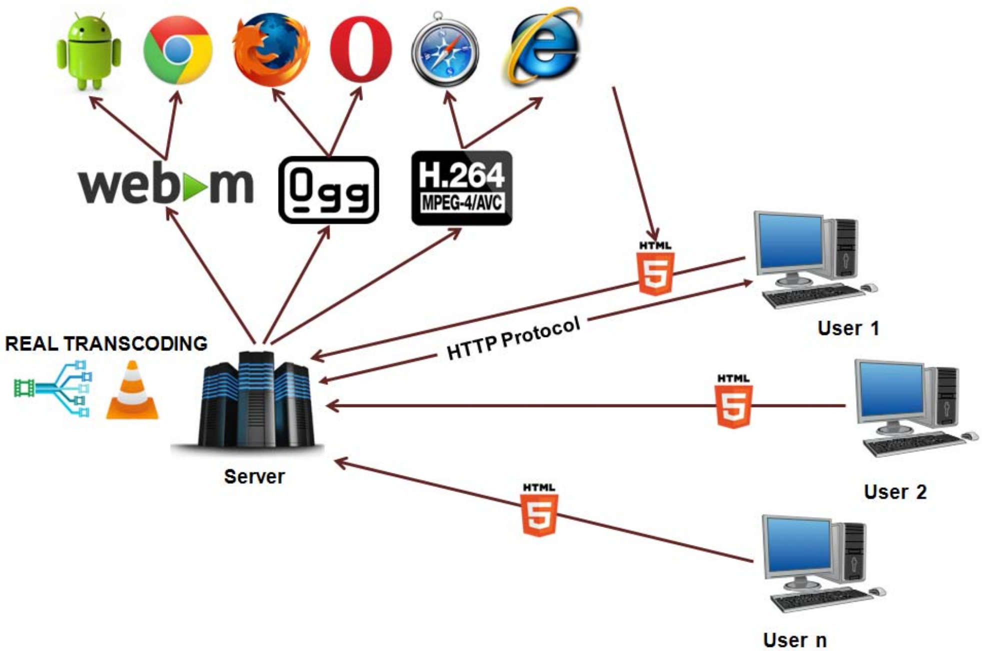


***Figure 5.51.*** *Overview of the system procedure when a user requests video.*

In order to transcode in real-time each second of video, it must be coded in less than one second (or one second at most). When a file is transcoded, the original file (source) is coded to another codec which can have the same or lower bitrate but with the same duration, chroma subsampling, and frame rate. Our purpose is to design an algorithm that takes into account the characteristics needed to be able to stream the video in real time. These are the required parameters for the end user:

- Bitrate of the end user. It may be lower than the source file.

- Frame rate of the end user. It depends on the geographic zone: for Phase Alternating Line (PAL) and coleur Séquentiel Couleur à Mémoire (SECAM) it is 25 fps (frames per second), while for National Television System Committee (NTSC) it is 29.976.
- Browser used. In order to do it automatically, we can detect the type of browser via JavaScript code.

### 5.6.2 Experimental process description

In this section we describe the steps carried out in the experimental process for the study of the video transcoding to deliver it using HTML5. This process is split in several stages that let us analyze all cases in terms of video quality, bandwidth consumption and required transcoding time. It let us decide which compression codecs should be used in each case and which cases should be distinguished from others in order to apply real transcoding.

#### *5.6.2.1 Identify the parameters to analyze*

The first step is to compare different compression codecs. This allows us to know which parameters obtain different values for different encoders. In this stage we observed that different fps, as well as bitrates, provide us different data size and encoding times.

We performed several tests with the purpose of searching the best video codec taking into account the final size of the video once it is compressed, the coding time and the final video quality. This analysis made us split all cases into four subgroups: size PAL, size NTSC, coding time PAL and coding time NTSC. Moreover, we assigned to these subgroups different coding bitrates: 64, 240, 576, 1152, 1536, 2256, 3024, 4224, 5808 and 7950 Kbps that correspond to the standard bitrate values with VLC software. We have included very high data transfer rates because some users are starting to use very high quality standards such as the Ultra High Definition 4K (3840x2160).

#### *5.6.2.2 Test the encoding time*

The main purpose of this stage is to test whether the time needed to encode the video to another codec is lower than the time playing it or not. If the time is higher, the user will not be able to play it in real time, because there will be one point that there will not be enough video transcoded to be played at the end user side. There should be an external counter synchronizing very precisely the execution start and end of the encoding.

#### *5.6.2.3 Study the compression ratio*

For each codec and each bitrate we measured the compression ratio. This test let us measure what type of codec and compression is more favorable for any case depending on whether the priority is the encoding time or the end of file size, as it may be the case of a broadcast over the Internet. The compression rate for the calculation of the ratio is defined as the ratio between the size of the raw video file and the compressed one.

#### *5.6.2.4 Parameters needed for transcoding*

After having performed a bulk of tests, in this subsection we discuss the parameters to be taken into account in the proposed algorithm in order to decide which codec should be used in the transcoding process for each case.

The first parameter is the type of browser. We observed that in the case of Internet Explorer, mainly used in Microsoft Operative System, and Safari, mainly used MAC OS, perform better the encoding using MP4 (H.264). We have observed that MP4 codec has higher compression rate than WebM and higher video quality than Ogg. Therefore, it seems to be the best candidate for real-time encoding in this case (although, WebM and Ogg can also be used because they have lower coding time than the playback time, so it can be transcoded in real time). Then, it is chosen the frame rate. It is important to choose between PAL and NTSC, because PAL has close to five fps less than NTSC, so it speeds up the encoding process as well as reduces the final video file size. When we measured the compression time for the bitrates from 64 to 7950 kbps, we have seen that that there is a turning point. Below this point, the coding time per second is lower than a second of playback. We should not forget the quality of the video. Increasing the bitrate we can find balance point between the compression time and the video quality. In terms of video quality, we had a scale with 5 values: very poor quality, poor quality, medium quality, good quality and very good quality. Then we tagged each bitrate with the appropriate value of this scale. Finally, the system only provides the options of good quality and very good quality. The system balances between the type of browser, the frame rate, the bitrate and the video quality, but taking into account that the coding time should be lower than one second per playback second. For 1536 Kbps to 7950 Kbps, with a PAL frame rate and MP4 compression, we have achieved an assessment from good to very good video quality.

### 5.6.3 Experimental results

In this section we discuss and compare the results obtained when the algorithm is running.

In Figure 5.52 we can see the trend of the graph as the bitrate grows in the X-axis. These bitrates range from 64 Kbps to 7950 Kbps. The turning point for MP4 is achieved at 1536 Kbps, which has 0.981 ms of encoding time per payback second. Then, we can only make real transcoding from 64 Kbps up to 1536 Kbps. We used VLC media player 2.1.3 RinceWind to get to the turning point of transcoding in real time. The turning point can be considered as the point where the balance between the quality and the compression time. Therefore, we guarantee at this point that lower encoding times are achieved in higher quality videos. We can see that from this turning point the file size is reduced by 50% with respect to the final file. From this point it increases almost exponentially.

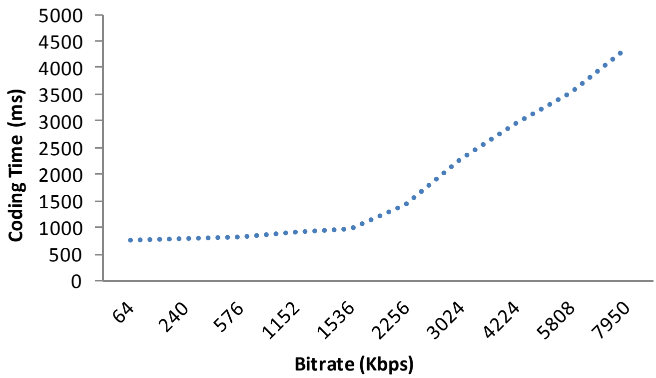


**Figure 5.52.** *Coding time according to different bitrates for MP4.*

In Figure 5.53, we see the trend of the graph when we vary the bitrates from 64 Kbps to 7950 Kbps for different compression codecs compatible with HTM5. All files had PAL frame rate. We used VLC media player 2.1.3 RinceWind to transcode them. MP4 is the codec which has higher compression (very close to Ogg). After 1152 Kbps, we can see that the size starts to be gradually more different (between MP4 and Ogg, the difference becomes more appreciated after 2256 Kbps).

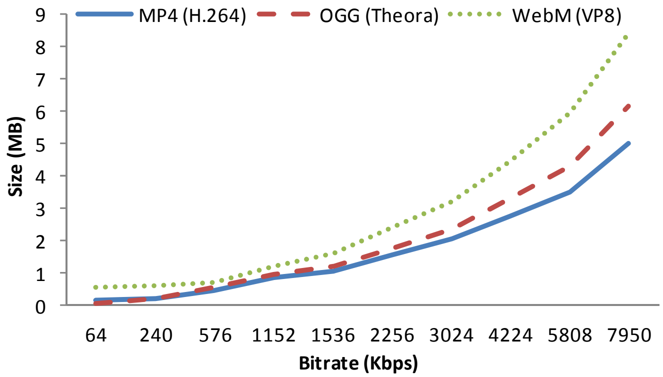


**Figure 5.53.** *File sizes in PAL for different bitrates.*

In Figure 5.54 we can see the trend of the graph MP4, Ogg and WebM file sizes, when their compression bitrates vary from 64 Kbps to 7950 Kbps. We used in all cases NTSC frame rate. The software used to perform this test was VLC media player 2.1.3 RinceWind. In this case we observed that MP4 has a turning point at 2256 Kbps, where we appreciated a change in the compression ratio (from 597:1 to 597:2, that is, from 1.18 MB to 1.70MB) with a sudden increase of 30 %.

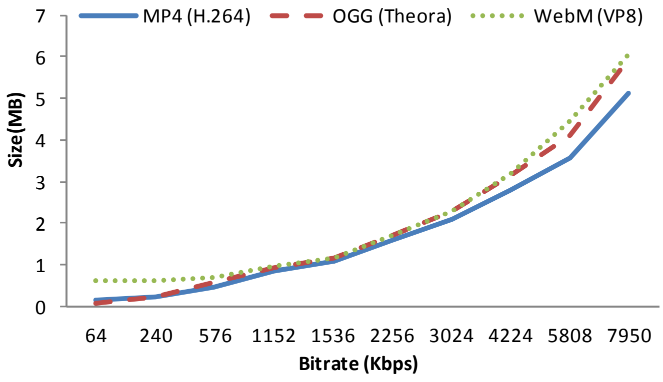


**Figure 5.54.** *File sizes in NTSC for different bitrates.*

In Figure 5.55, we can see the trend of the graph by applying different bitrates from 64 Kbps to 7950 Kbps for MP4, Ogg and WebM when measuring the coding time. In this case we used PAL frame rate and the used software was VLC media player 2.1.3 RinceWind. The results obtained for the coding time are very different. WebM here is positioned as a more balanced codec taking into account the compression time. Ogg becomes the option that needs more time and, moreover, it does not provide better quality results than WebM. The better feature of MP4 is its balance between the video quality, the final file size and end time, that is why it is chosen by many researchers.

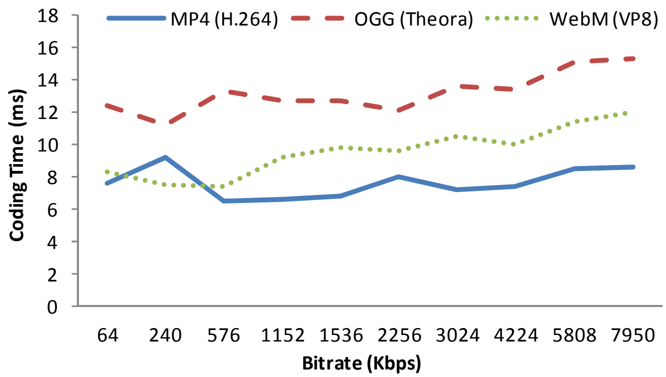


**Figure 5.55.** *Coding time for PAL for different bitrates.*

In Figure 5.56 we can see the trend of the graph by applying different bitrates (from 64 Kbps to 7950 Kbps) to MP4, Ogg and WebM to estimate the coding time. In this case we used NTSC frame rate when using VLC media player 2.1.3 RinceWind.

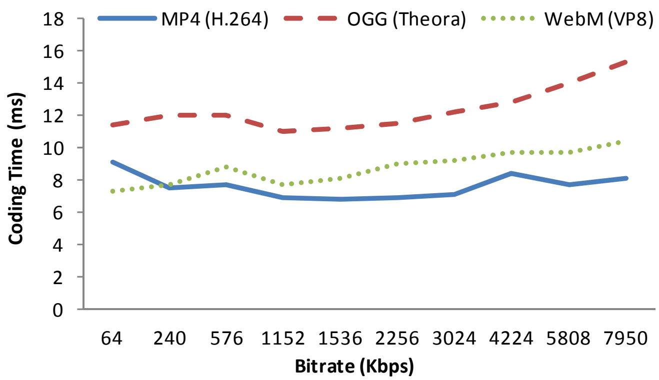


**Figure 5.56.** *Coding time for NTSC for different bitrates.*

After having performed all these tests, we can provide the following discussions. We observed that the codec that provided the smaller size always was MP4 in the PAL and NTSC systems. WebM had the bigger size for both PAL and NTSC system. The one that provide lowest coding time was always MP4 (very close to WebM) in both PAL and NTSC systems. Finally, the best quality corresponded to WebM while lowest quality to MP4 in all of the options. Therefore, MP4 (H.264) offers better coding times with very low file size. It is just the opposite of WebM, which increases its coding time and the end file size in favor of an excellent quality. Ogg (Theora) has been in between both codecs in all cases. In terms of video quality, the best compression codec for streaming video of HTML5 has been WebM. Table 5.12 shows a summary of the results obtained in reference to the Quality of Experience:

**Table 5.12.** *Better codec in terms of QoE as a function of the parameters.*

| **Parameters** | **Formats and Software** | | | |
|---|---|---|---|---|
| | **PAL VLC** | **NTSC VLC** | **PAL SUPER** | **NTSC SUPER** |
| **< Size** | MP4 | MP4 | WebM | WebM |
| **> Size** | WebM | WebM | OGG | OGG |
| **< Time coding** | MP4 | MP4 | MP4 | MP4 |
| **> Time coding** | OGG | OGG | WebM | WebM |
| **< Quality** | MP4 | MP4 | MP4 | MP4 |
| **> Quality** | WebM | WebM | WebM | WebM |

## 5.7 Test for choosing the Best Video Compression Codec

In order to check whether the compression of a video depends on the predominant color of the video or not, we have generated videos with different characteristics. Each video has a background with one dominant color and a cube of the same color (but darker) that appears on the screen bouncing on the four sides. The original videos have one minute duration with 25 fps, with a file size of 8.68 GB. The aspect ratio of each video is 16:9. The image size of the videos is 1920 x 1080 pixels. The bit rate of the original videos is 1244 Kbps. These videos are compressed using the 4 codecs under study (DivX, XviD, MS MPEG4-v2 and H.264 part 10).

The first one is a video with black background and a black cube in the middle of the picture. Figure 5.57 shows the histogram of video with black as predominant color. It is denoted because red, green and blue peaks appear at the left side of the histogram.

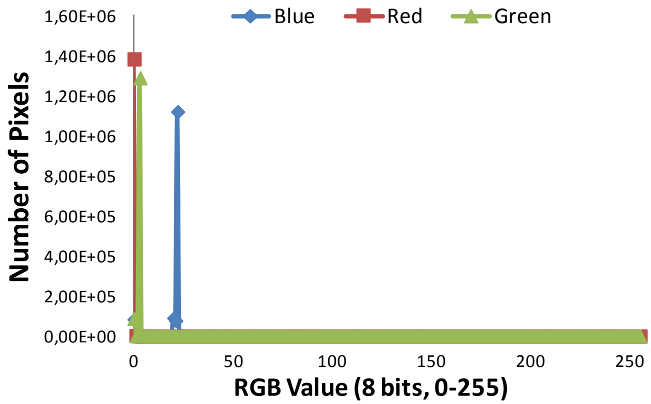


**Figure 5.57.** *Histogram of the video with black as predominant color.*

Second video shows a red background and a red cube in the middle of the picture. Figure 5.58 shows the histogram of video with red as predominant color. It is denoted because the red peaks appear at the right side, while green and blue peaks (given because it was not a pure red color) are in the center part.

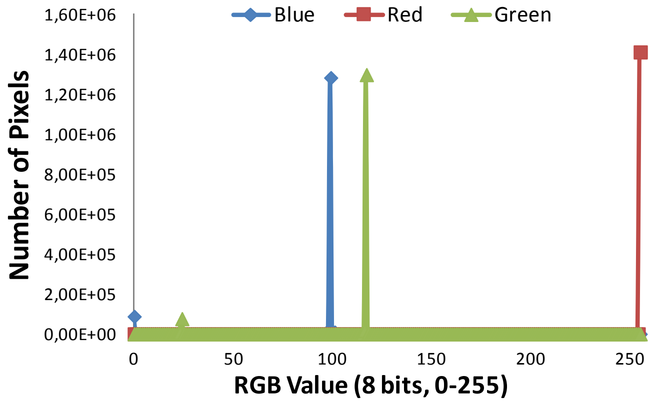


**Figure 5.58.** *Histogram of the video with red as predominant color.*

Third video shows a green background and a green cube in the middle of the picture. Figure 5.59 shows the histogram of video with green as predominant color. It is denoted because the green peaks appear at the right, while red and blue (given because it was not a pure green color) peaks are in the center part.

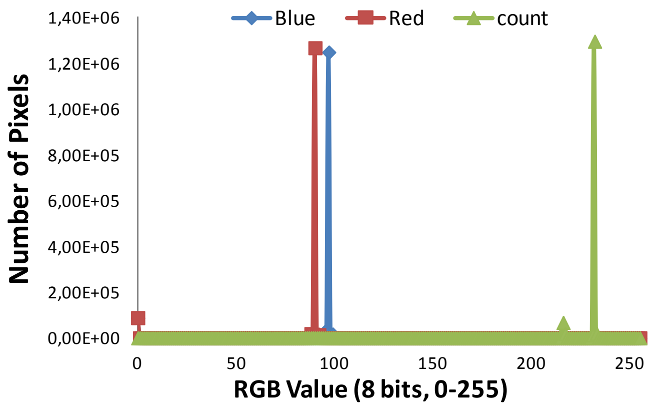


**Figure 5.59.** *Histogram of the video with green as predominant color.*

The fourth video shows a blue background with a blue cube in the middle of the image. Figure 5.60 shows the histogram of video with blue as predominant color. It is denoted because the blue peaks appear at the right side, while red and green (given because it was not a pure green color) peaks are in the left part of the histogram.

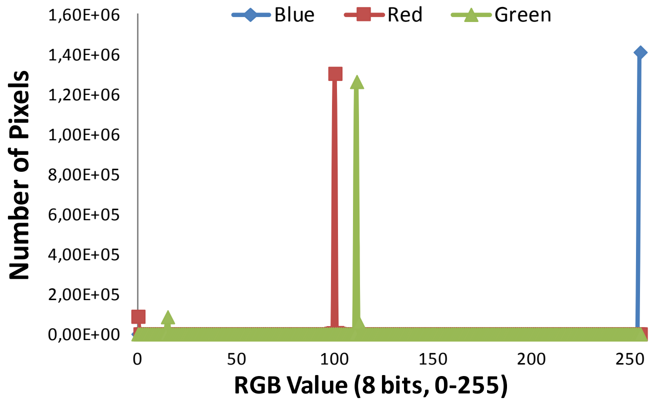


**Figure 5.60.** *Histogram of the video with blue as predominant color.*

Last video presents a white background with a white cube in the middle of the picture. Figure 5.61 shows the histogram of video with white as predominant color. It is denoted because red, green and blue peaks appear at the right side of the histogram.

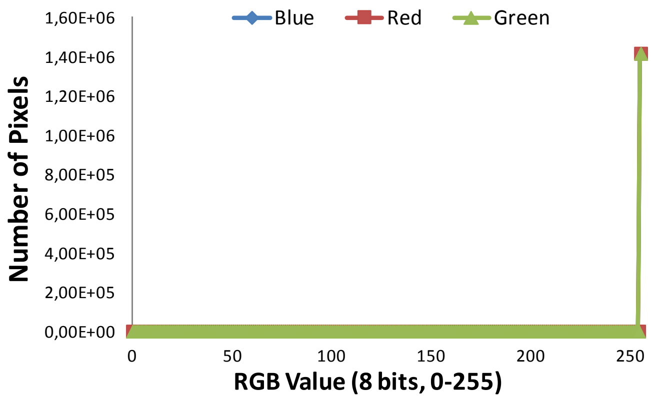


**Figure 5.61.** *Histogram of the video with white as predominant color*.

The result of compression for each case is a video with the same aspect ratio and image size, but with a lower file size. Moreover, we have decreased the bitrate. In this case, all compressed videos have a bit rate of 1008 Kbps. Final sizes for each video file, classified by colors, are shown in Figure 5.62. When we compare colors, we can see that black and white colors suffer higher compression than the rest. If we pay attention to red, green and blue colors, we can see that in DivX and MS MPEG4_v2 there is higher compression for green than for red and blue, while for H.264 Part 10 and XviD there is higher compression in red and blue than green.

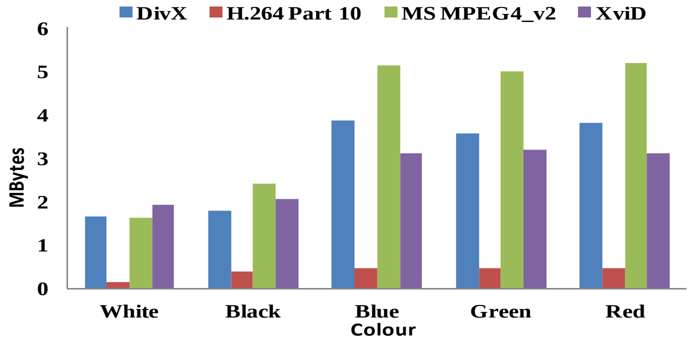


**Figure 5.62.** *Final video size after the compression as a function of the color.*

If we compare the result of compressing videos taking into account the codecs (Figures 5.63 and 5.64), we can see that videos with black and white backgrounds suffer greater compression using DivX codec than with XviD codec. However, videos obtain lower size when compressing with XviD than with DivX when using red, green and blue. H.264 part 10 is the codec that provides the highest level of compression for all cases and MS MPEG4_v2 is the worst for black, red, green and blue.

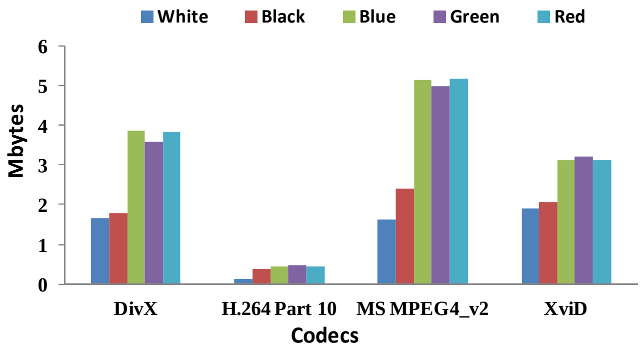


**Figure 5.63.** *Final video size after the compression as a function of the codec.*

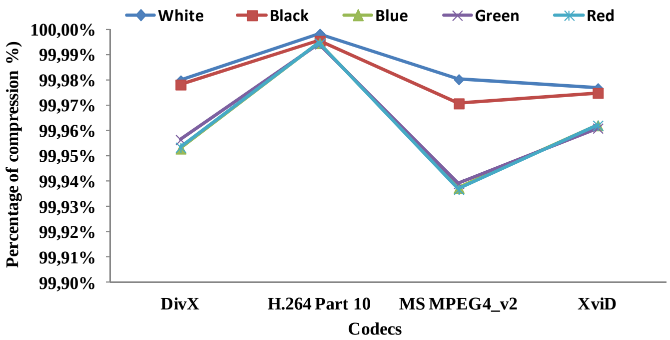


**Figure 5.64.** *Compression percentage of in (°/1) for all videos as a funtion of the used codec.*

Additionally, we have performed a comparison between the different codecs using documentary real videos, and focusing on image quality and compression time using documentary real videos. These are some of the important factors taken into account when selecting the videos. To perform our test, we selected 10 seconds (250 photograms) from 4 videos. The first one is a short video of the mouth of a river through a forest. In this video the predominant color is the green of the trees, see Figure 5.65.

**Figure 5.65.** *Video with green as predominant color.*

The second video, see figure 5.66, presents a scene of a desert where there are some persons sat over the sand. These persons generate shadows on the ground. In this video we observed the red as the predominant color.

**Figure 5.66.** *Video with yellow as predominant color.*

The third video presents a scene in Artic where the glacier in movement is shown. In this case, the predominant color is white,see Figure 5.67.

**Figure 5.67.** *Video with white as predominant color.*

Last video, see Figure 5.68, is a scene where there is a fishing net with fishes swimming. This scene presents blue as predominant colors.

**Figure 5.68.** *Video with blue as predominant color.*

Table 5.13 shows the results summary of image quality and time compression for the videos shown in Figures 5.65, 5.66, 5.67 and 5.68. Because the compression time also depends on the personal computer hardware characteristics and on the used software, we have relatively compared them giving Low, Medium and high values. For video quality, we compared them using: Medium-Low Quality, High-Medium Quality, Medium Quality and Worst Quality.

**Table 5.13.** *Results summary of image quality and time compression.*

| | **Blue** | | **Green** | | **Red** | | **White** | |
|---|---|---|---|---|---|---|---|---|
| **Codec** | **Time Compres.** | **Image Quality** | **Time Compres.** | **Image Quality** | **Time Compres.** | **Image Quality** | **Time Compres.** | **Image Quality** |
| **Original** | - | Best Quality | - | Best Quality | - | Best Quality | - | Best Quality |
| **MPEG-4 Microsoft** | Low | Medium-Low Quality | low | Medium-Low Quality | Low | Medium-Low Quality | Low | Medium-Low Quality |
| **XViD** | Low | High-Medium Quality | Medium | High-Medium Quality | Medium | High-Medium Quality | Medium | Medium Quality |
| **DivX** | Medium | Medium Quality | Medium | Medium Quality | Medium | Medium Quality | Medium | High-Medium Quality |
| **H.264** | High | Worst | High | Worst | High | Worst | High | Worst |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Quality | | Quality | | Quality | | Quality |

# 5.8 Selection of Codecs in video surveillance

To acquire the sequence of images, which we will later study, we used the Sony DCR-IP5 camera, whose video acquisition format is in DV and has a resolution of 720x576 in the PAL system. The plane where the sequence of images has been captured is an exterior with a large content of brown and blue. Although the background of the image is fixed, there is movement due to the passage of cars on the road. Figure 5.69 shows one of the images of the place.


**Figure 5.69.** *Image of the capture site.*

## 5.8.1 Test bench and Results

The camera has been connected to a computer through an IEEE 1394 interface. The computer used for the capture is a 1300 MHz Intel

Celeron-S with 320 Mb of RAM. The operating system used was Microsoft Windows XP Professional ®. The application to code that has been used is VirtualDub version 1.6.2 (free code). For the compression, DivX codecs version 5.2.1, XviD 1.0.3-20122004 (Koepi) and H.264 version x264vfw_rev120 have been used. The program to collect measures is the Microsoft 2.0 Management Console ©. This application allows us to measure the performance of the computer while the VirtualDub process works. There was no other process running on the computer while compression was running. The application started manually when video encoding was started and stopped manually at the end of the process. The encoding of the different videos with different bit rates has been done without audio, without any filter and in the "Fast Recompress" mode of VirtualDub. In the 3 codecs we have selected the coding of a single pass (faster coding), because the two pass is slower and is not useful for transmitting image sequences.

To obtain measurements, video has been captured at 9:00 and 12:00 in the morning and at 5:50, 18:20 and 19:30 in the afternoon. These hours of the day have different light and therefore the images have different brightness, in addition, there is a different level of traffic, and therefore different movement. The sequences of images have been compressed for 5 types of bit rates: 909, 1649, 2389, 3129 and 3869.

According to the results obtained for the compression of the videos taken at 12:00 and 19:30 hours (Figures 5.70 and 5.71), it can be seen that of the 3 codecs used, the one with the highest compression factor is the XviD.

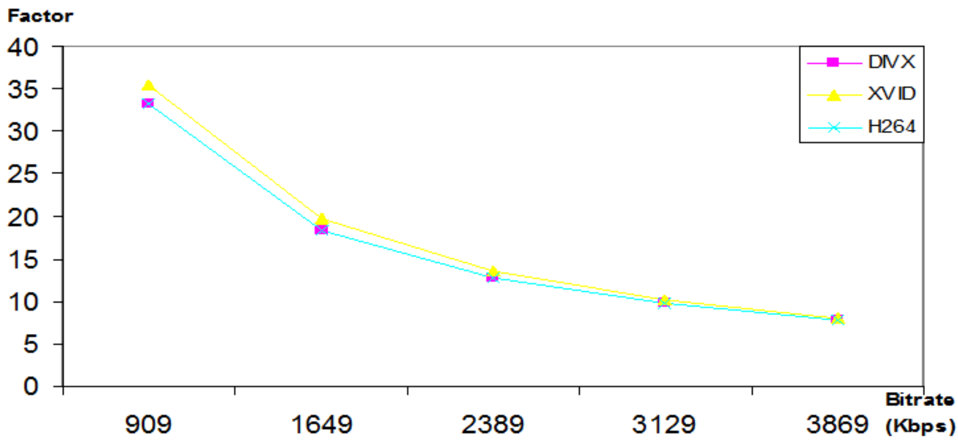


**Figure 5.70.** *Compression factor (12:00).*

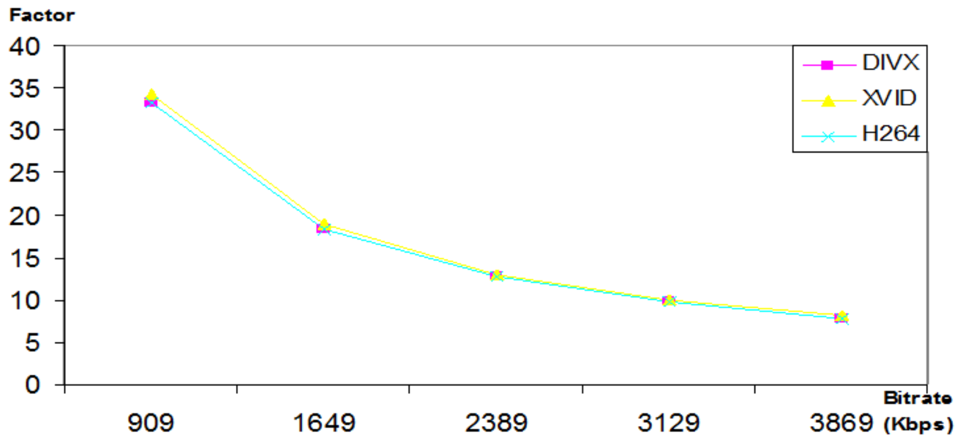


**Figure 5.73.** *Compression factor (19:00).*

This difference becomes more noticeable when there is less luminosity (at night) in the images to be compressed. Therefore, when the limitation in a video surveillance system is imposed by the bandwidth of the transmission channel, either because there are many cameras or the channel has low bandwidth, the best codec to use of the three analyzed is XviD.

Comparing the time it takes to compress a second video of the acquisitions made at 12:00 and 19:30, Figures 5.72 and 5.73 respectively, we realize that the more darkness there is, the more time it takes to compress a second of video. Therefore, the higher the compression bit rate, the more time it takes to encode that second. This difference becomes more acute for H. 264. The codec that compresses faster is DivX, although the difference with XviD is very short. DivX and XviD are the best when short compression times are needed.

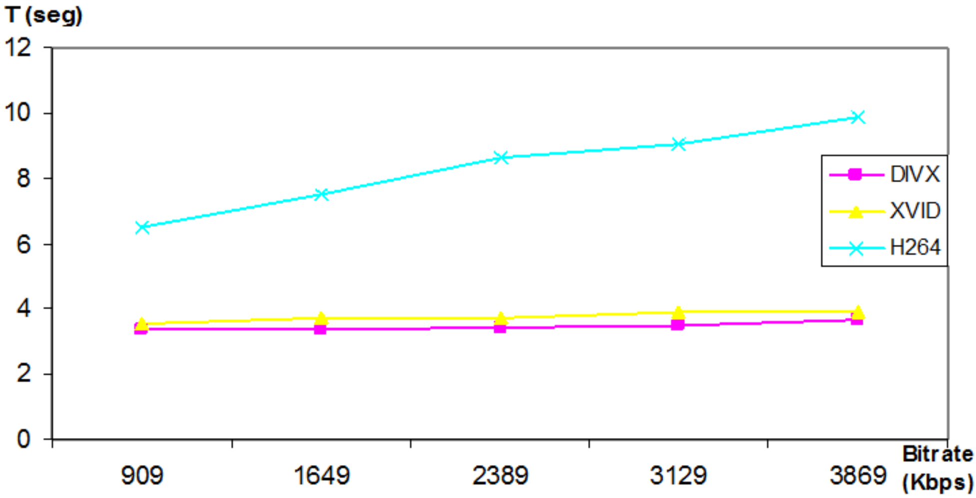


**Figure 5.72.** *Time to compress a second of video (12:00).*

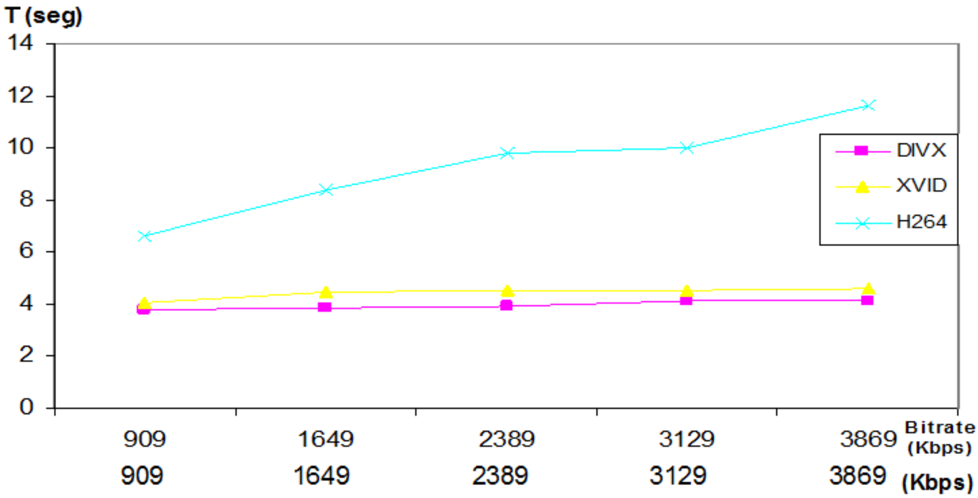


**Figure 5.73.** *Time to compress a second of video (19:00).*

For a certain sequence of images, if we use a higher bit rate, the microprocessor takes longer to reach 100% of the processor time, as can be seen in Figures 5.74 and 5.75. This happens with all videos regardless of the luminance they have. The DivX codec suffers most from this effect. If the limitation of the system that we implement, is in the processing speed of the microprocessors of the equipment that is being used, it is better to use small bit rates.

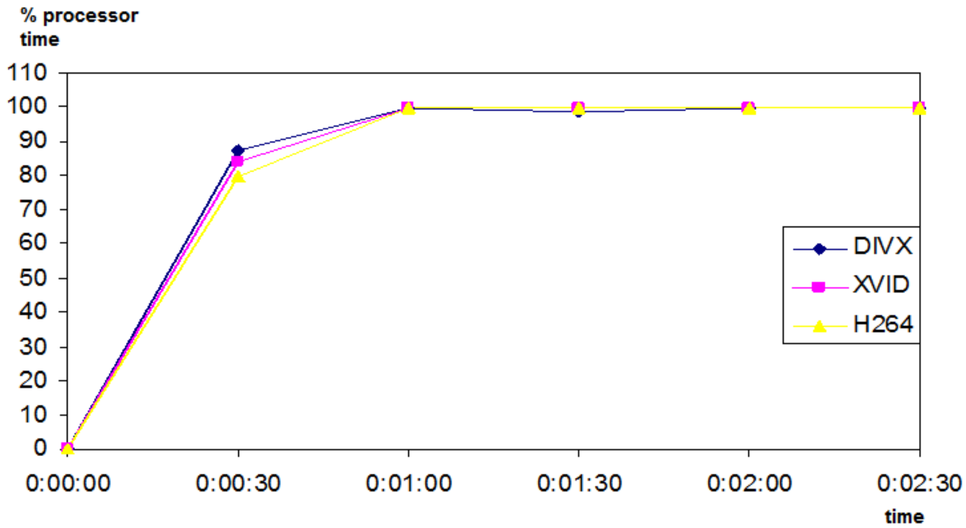


**Figure 5.74.** *% time of processor, video 909 (9:00).*

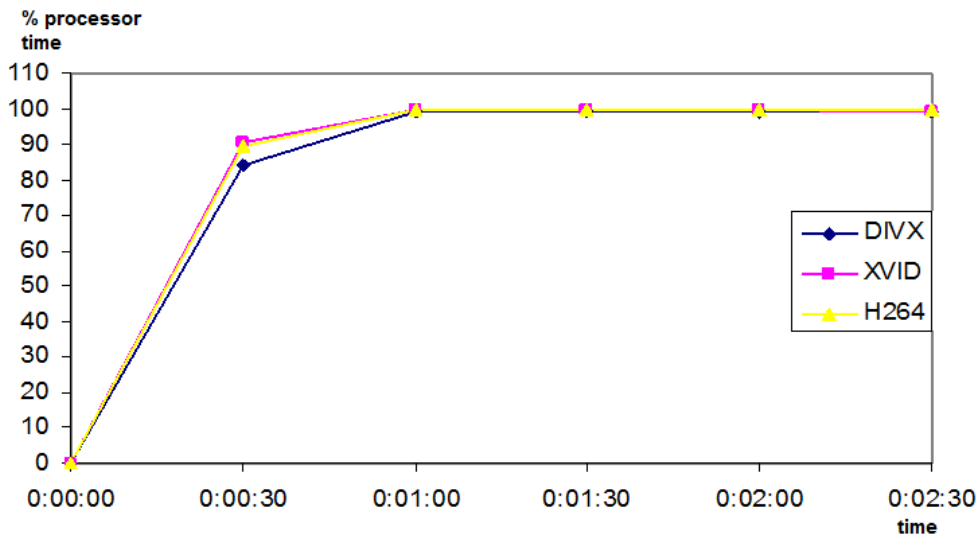


**Figure 5.75.** *% time of processor, video 3869 (9:00).*

Comparing the operations of Input/Output in a sequence of images captured at 17:50, with different bit rates (909 and 3869), as shown in Figure 5.76, we realize that, at a lower bit rate, there is more Input/Output data during the start of compression. However, the results obtained for different bit rates in the video at 19:30 hours, Figure 5.77, show that at a higher bit rate there is more Input/Output data during startup. Figure 5.78 shows the video that corresponds to 18:20 hours. For both a low and high bit rate, the Input/Output data is almost equal. Therefore, the number of Input/Output operations when the compression is performed depends both on the bit rate used and the luminance of the images to be compressed. The codec with the least number of Input/Output operations is H.264.

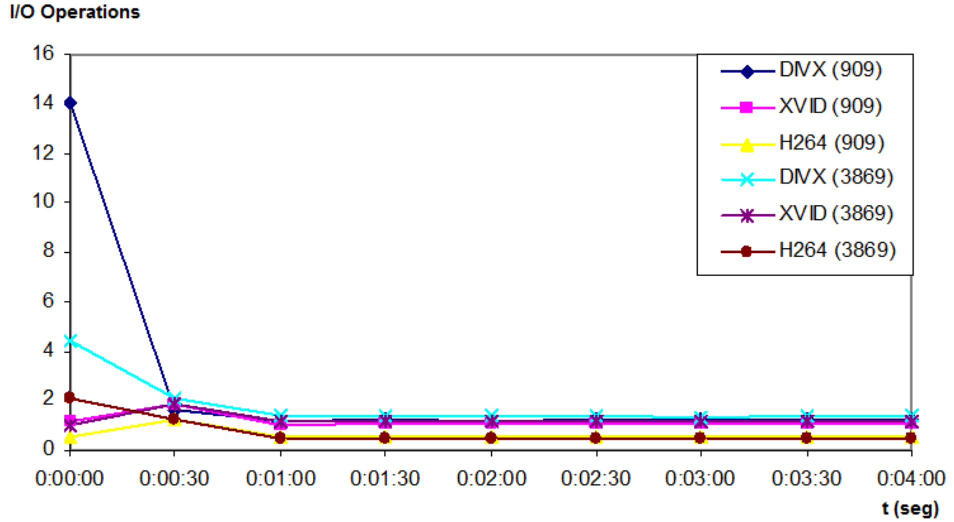


**Figure 5.76.** *I/O operations per second at 17:50.*

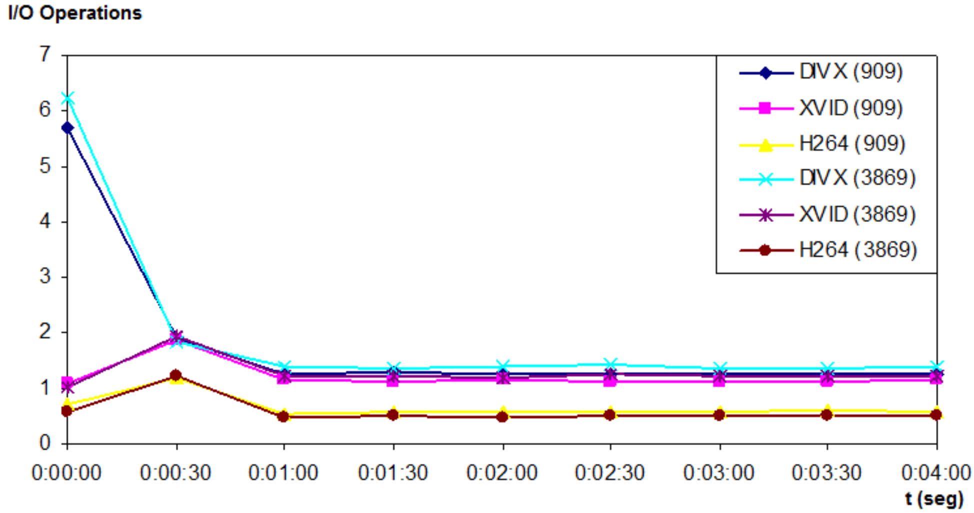


**Figure 5.77.** *I/O operations per second at 19:30.*

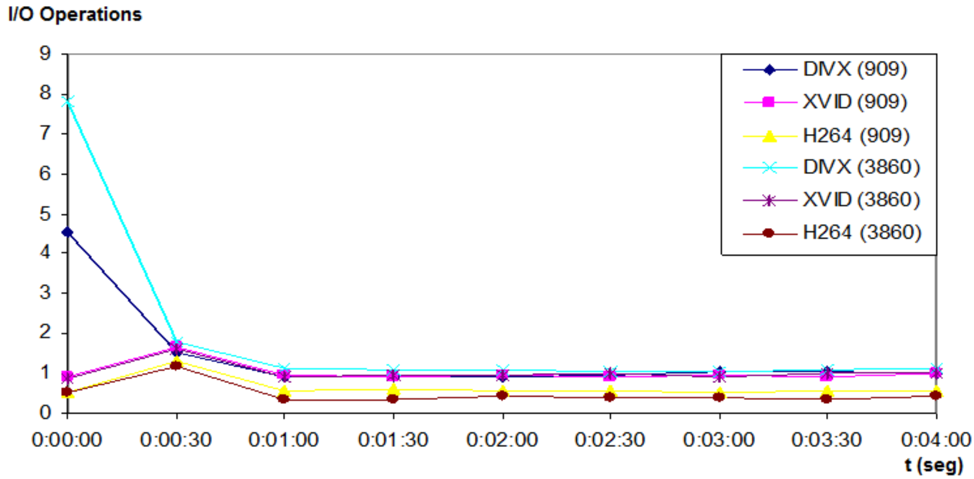


**Figure 5.78.** *I/O operations per second at 18:20.*

## 5.9 Performance test of autonomous video compression system for environmental monitoring

The system is tested with different videos and the network performance is measured in terms of consumed bandwidth, packet loss, jitter and delay.

The system is based on a server that executes our algorithm developed in Python and MATLAB® and analyzing the network constraints and the predominant color of videos provided by cameras. The algorithm analyzes the RED-GREEN-BLUE (RGB) components of the video and performs the transcoding tasks.

Figure 5.79 shows the process carried out to carry out the preliminary study for the characterization of each of the resulting videos and the computation time to recode the videos.

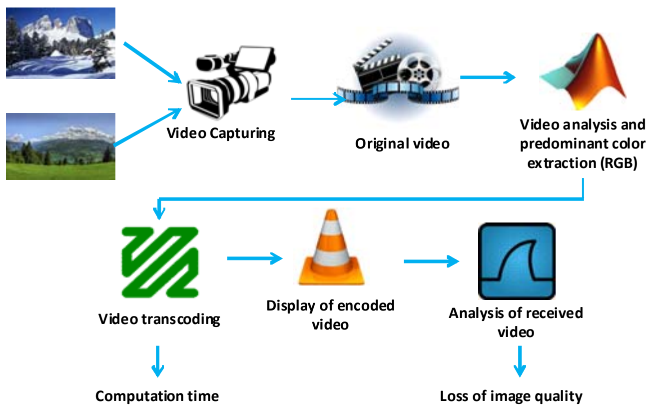


**Figure 5.79.** *Description of video transcoding process.*

In order to determine which codec best fits a video recorded in a natural environment according to its average RGB level and its predominant color, a short algorithm has been developed that quickly and efficiently determines which ones are the color parameters of a video desired for this study. To do this, develop our algorithm, MATLAB® has been used. MATLAB® consists of very powerful and specific tools that allow performing all kinds of data processing of various natures. In this case, for the development of the present study, one of the integrated applications of image processing, Video Viewer, and the native functions of the software will be used. Through this integrated application, it is possible to perform a multitude of tasks, among which are those that in this work concern: viewing a video, accessing relevant information, such as the exact duration, the frame rate at which it has been recorded, the duration of the video; and the visualization, frame by frame, of the complete pixel map, thus being able to check which is the corresponding RGB parameter of each of the pixels of each video frame in a very simple way. Once the tools that will form the basis of the algorithm are detailed, the description of it is presented before in section 3.2.4.1).

Thus, the result of said study has resulted in the following values shown in table 5.14

**Table 5.14.** Video processing result by RGB algorithm implemented in MATLAB®.

| **Original Video File** | **RGB Levels** | | |
|---|---|---|---|
| | ***Red (R)*** | ***Green (G)*** | ***Blue (B)*** |

| Original Video File | RGB Levels | | |
|---|---|---|---|
| | *Red (R)* | *Green (G)* | *Blue (B)* |
| azul.mp4 | 37 | 117 | **154** |
| whales.mp4 | 24 | 70 | **149** |
| rojo.mp4 | **127** | 79 | 51 |
| eleph.mp4 | **131** | 91 | 58 |
| verde.mp4 | 67 | **93** | 64 |
| dogs.mp4 | 89 | **97** | 55 |
| black.mp4 | **44** | **44** | **44** |
| white.mp4 | **186** | **195** | **210** |

As can be seen in the exposed table 5.14, each of the original video files has been processed using the RGB algorithm developed. For each one of them, the predominant chroma has been highlighted, with the exception of the videos whose chromatic spectrum turns out to be white or black, since these two chromas are characterized by:

• White chroma: its three RGB levels have a value very close to 255 and a very small difference between them.

• Black chroma: its three RGB levels have a value very close to 0 and a very small difference between them.

### 5.9.1 Considerations related to codec compatibility

We must know that not all video formats (or containers) are compatible with all codecs. So a small study based on tests has been carried out in FFmpeg and VLC to check with which codecs the containers used in this work are compatible.

Table 5.15 shows the compatibilities of video formats and codecs to which we can convert each video:

**Table 5.15**. *Relationship between containers and compatible codecs.*

| Original Video Format | Compatibility of Codecs | | | | | |
|---|---|---|---|---|---|---|
| | *APCN* | *FMP4* | *H264* | *WMV1* | *FLV1* | *XVID* |
| .mp4 | × | ✓ | ✓ | × | × | × |
| .avi | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

### 5.9.2 Test bench and Results

First, the results obtained for each of the studies or analyzes carried out will be presented. In these will be collected the first conclusions on the decision that would be taken in each case regarding the choice of the compression codec. Secondly, the conclusions obtained in the previous section will be collected and the decision algorithm will be designed. To carry out we have used 2 computers connected wirelessly, see Figure 5.80, consists of 3 devices: a personal computer that will act as a stream server, a second personal computer, which will act as a client of the stream and a WIFI router.

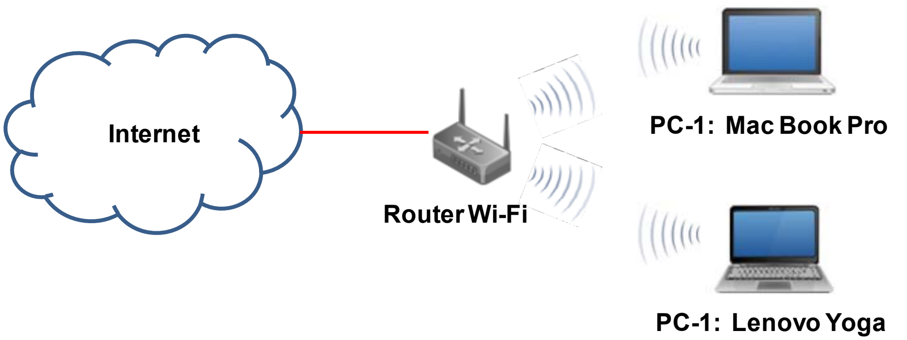


**Figure 5.80.** *Proposed scenario for transmitting the video.*

#### *5.9.2.1 Results of FFmpeg*

The objective of this study is to determine, given an input video in a given codec, the time it takes to code to one of the codecs that have been established as an object of study in this project, as well as the obtaining of said video files recoded.

As a step prior to the recoding of the files, a change has been made from the original video container to the .avi container, since it is the container format that offers the most compatibility when it comes to recoding.

After performing the recodifications corresponding to those shown in the previous table 5.15 for each of the videos, by using the time command in each sentence, the user, system and real time involved in performing said processes are obtained.

Prior to the presentation of the temporal results, an analysis of how the size of the video files varies by applying a certain compression codec has been made. For this, the file size data in the .avi container format have been observed and compared with the resulting size in the same container format for each of the video codecs under study.

In the presentation of results of this parameter those corresponding to the APCN codec (Apple ProRes) have been omitted because, far from reducing the size of the original file, when recoding it with this codec it increases in a percentage relation sometimes higher than 1000%. Figure 5.90 shows the compression ratio for each case.

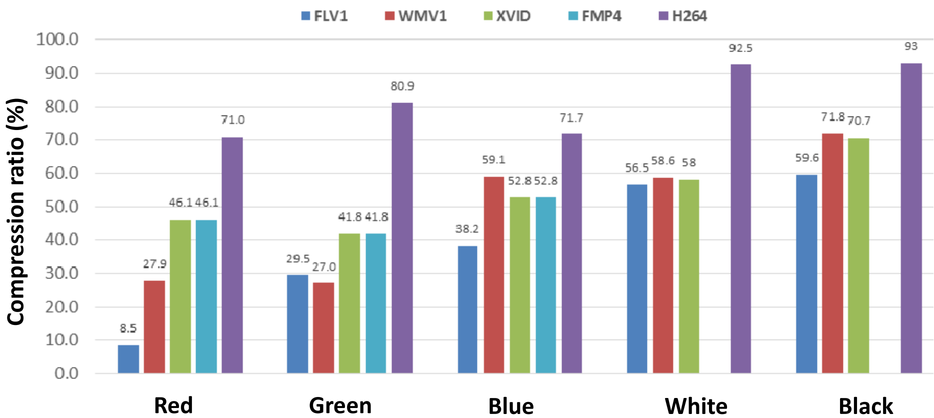


**Figure 5.90.** *Proposed scenario for transmitting the video Percentage of reduction of the size of the video file with respect to the original according to the compression codec applied to a video with a certain RGB.*

As can be seen in Figure 5.91, in general, the compression codec H264 offers a percentage reduction in size greater than the rest of codecs independently of the chromatic spectrum present in the video.

This result is the first of the parameters that will be taken into account when making a decision algorithm to establish which compression codec is most suitable according to a predominant color throughout a video.

By means of the real time parameter that is used in recoding a video, a deduction can be made of the video codec that offers a more adequate solution for the result of the time used to perform its compression.

In the table 5.16, shown below, the results obtained by the time parameter of the FFmpeg command are collected for each of the videos under study (except for the so-called 'red', 'blue', and 'green'), since the study of these was not relevant because its duration was around 1s of duration and the result obtained from compression time was not going to be relevant), according to the input codec and taking into account the table' Video processing result by RGB algorithm implemented in MATLAB®.

***Table 5.16.*** *Results of the elapsed time in the execution of the FFmpeg command.*

| **Video Description** | **Results of the elapsed time** | | | |
|---|---|---|---|---|
| | **Output Codec** | **Real time (s)** | **User time (s)** | **System time (s)** |
| Whales (Blue as Predominant color) | APCN | 1.770 | 6.817 | 0.005 |
| | FMP4 | 0.331 | 1.005 | 0.013 |
| | FLV1 | 0.351 | 0.773 | 0.009 |
| | WMV1 | 0.355 | 0.793 | 0.007 |
| | XVID | 0.334 | 0.966 | 0.013 |
| | H264 | 2.427 | 9.018 | 0.047 |
| Dogs (Green as Predominant color) | APCN | 1.203 | 4.638 | 0.035 |
| | FMP4 | 0.242 | 0.738 | 0.010 |
| | FLV1 | 0.398 | 0.582 | 0.008 |
| | WMV1 | 0.293 | 0.585 | 0.006 |
| | XVID | 0.241 | 0.744 | 0.010 |
| | H264 | 1.700 | 6.273 | 0.035 |
| Eleph (Red as Predominant color) | APCN | 1.166 | 4.506 | 0.031 |
| | FMP4 | 0.241 | 0.765 | 0.081 |
| | FLV1 | 0.280 | 0.587 | 0.007 |

| | | | | |
|---|---|---|---|---|
| | WMV1 | 0.300 | 0.594 | 0.006 |
| | XVID | 0.252 | 0.755 | 0.012 |
| | H264 | 1.820 | 6.737 | 0.036 |
| Black (Black as Predominant color) | APCN | 0.259 | 0.927 | 0.180 |
| | H264 | 0.147 | 0.434 | 0.023 |
| | FLV1 | 0.055 | 0.067 | 0.005 |
| | WMV1 | 0.062 | 0.074 | 0.005 |
| | XVID | 0.049 | 0.106 | 0.008 |
| | FMP4 | 0.244 | 0.400 | 0.017 |
| White (White as Predominant color) | APCN | 0.535 | 1.927 | 0.036 |
| | H264 | 0.323 | 0.905 | 0.056 |
| | FLV1 | 0.135 | 0.170 | 0.010 |
| | WMV1 | 0.136 | 0.171 | 0.009 |
| | XVID | 0.114 | 0.257 | 0.016 |
| | FMP4 | 1.054 | 0.157 | 0.010 |

As already mentioned, the temporary parameter that is going to be analyzed is the one called 'Real' and the justification offered is that it is the temporary parameter that explicitly measures the clock time it takes to execute the command from the moment it is entered.

That ends without taking into account internal processes of the CPU or processes that are running in other users of the system, as is the case of 'User time' and 'System time'.

In Figure 5.91 the results obtained for the parameter studied can be observed:

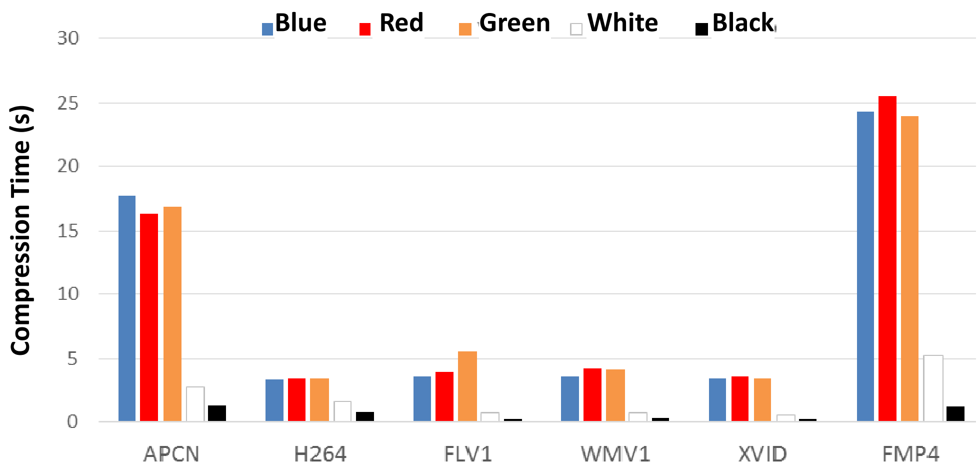


**Figure 5.91.** *Compression time.*

This result constitutes the second of the parameters that will be taken into account when making a decision algorithm to establish which compression codec is most suitable according to a predominant color throughout a video.

### *5.9.2.2 Network performace results*

The objective of the study carried out with the VLC and Wireshark software, tools for reproducing and analyzing multimedia files and for analyzing and capturing network traffic, is to obtain QoS (Quality of Service) parameters, which allow determining which codec is more suitable according to the results offered in its transmission. For this, the study described in said chapter has been carried out for four parameters, the bandwidth occupied in transmission, the packet loss experienced, the latency and the jitter.

### *5.9.2.3 Consumed bandwidth*

The average value of bandwidths consumed by the transmission of each video through the network are shown in Figure 5.92.

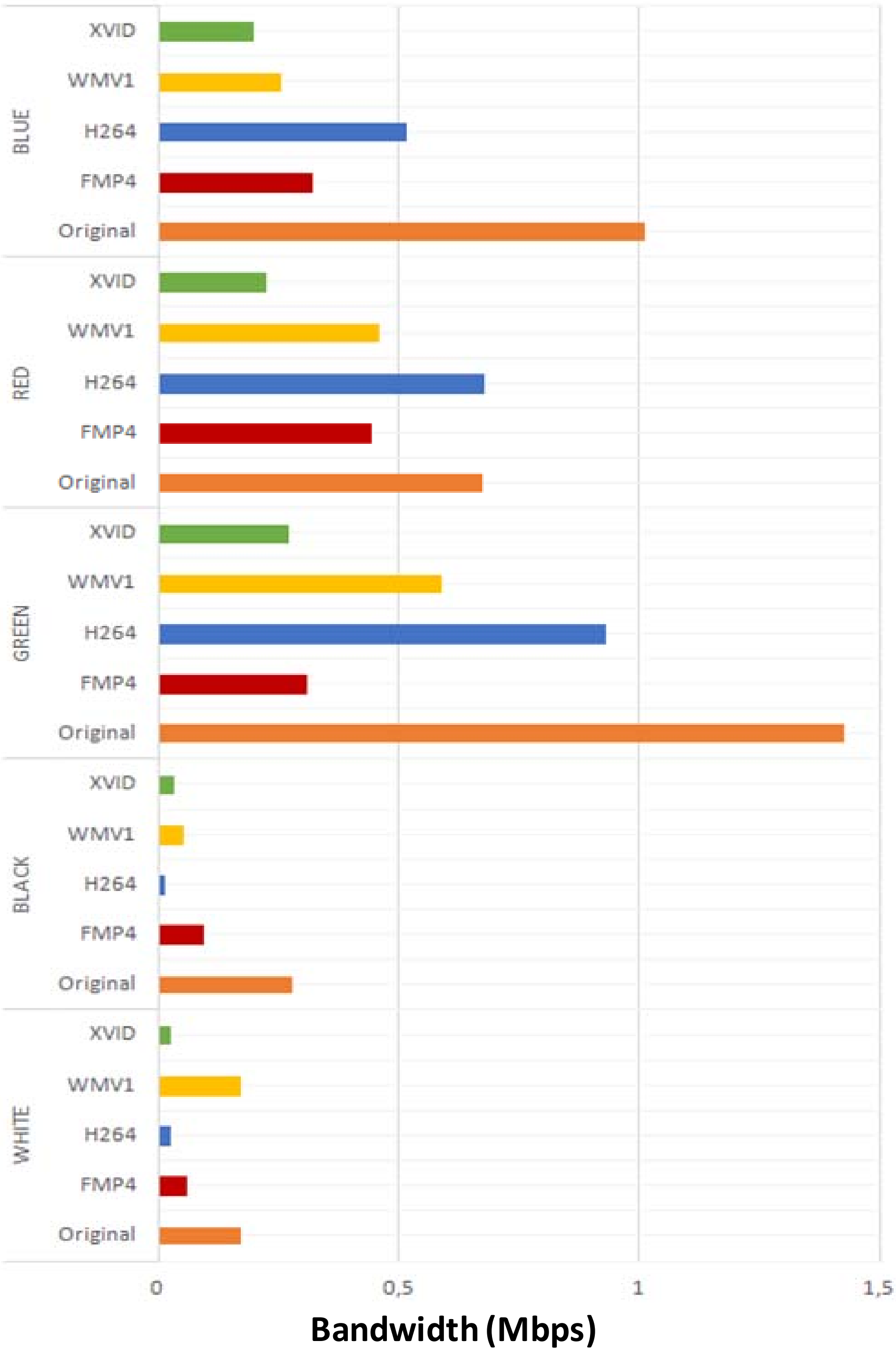


***Figure 5.92.*** *Average consumed bandwidth.*

As can be seen in Figure 5.92, the occupied bandwidth varies considerably between a chromatic spectrum video file and another regardless of the type of recoding performed.

It is possible to emphasize that, in the case of having a video in which the predominant colors are white or black, the H264 codec offers a remarkable result in terms of the occupied bandwidth.

In the case of having a video file in which the predominant colors were red, green or blue, the XVID codec offers the best result by providing a very low occupied bandwidth.

#### *5.9.2.4 Results of delay*

As with the previous parameter, for the presentation of results obtained regarding the latency observed in the transmission of a video, a comparative table has been prepared, and from it, a graph, to facilitate the interpretation of results comparing both the predominant color as the codec used in compression.

In the following Figures 5.93 and 5.94 we can see the comparisons made in terms of the average delay, see Figure 5.93, and the maximum delay, see Figure 5.94, obtained for each codec and the maximum delay according to the predominant color.

As can be seen in the graph of the average delay experienced, see Figure 5.93, for the files whose predominant colors are red, green or blue, the codec that offers the best results is H264, being a good result also offered by the XVID codec . If the maximum delay is also taken into account, see Figure 5.94, for the case of the predominant blue color, the XVID codec would be the best option.

For videos that have the predominant color white or black, both the average delay and the maximum delay, both are lower in the case of the XVID codec.

These results constitute the fifth of the parameters that will be taken into account when making a decision algorithm to establish which compression codec is most appropriate according to a predominant color throughout a video.

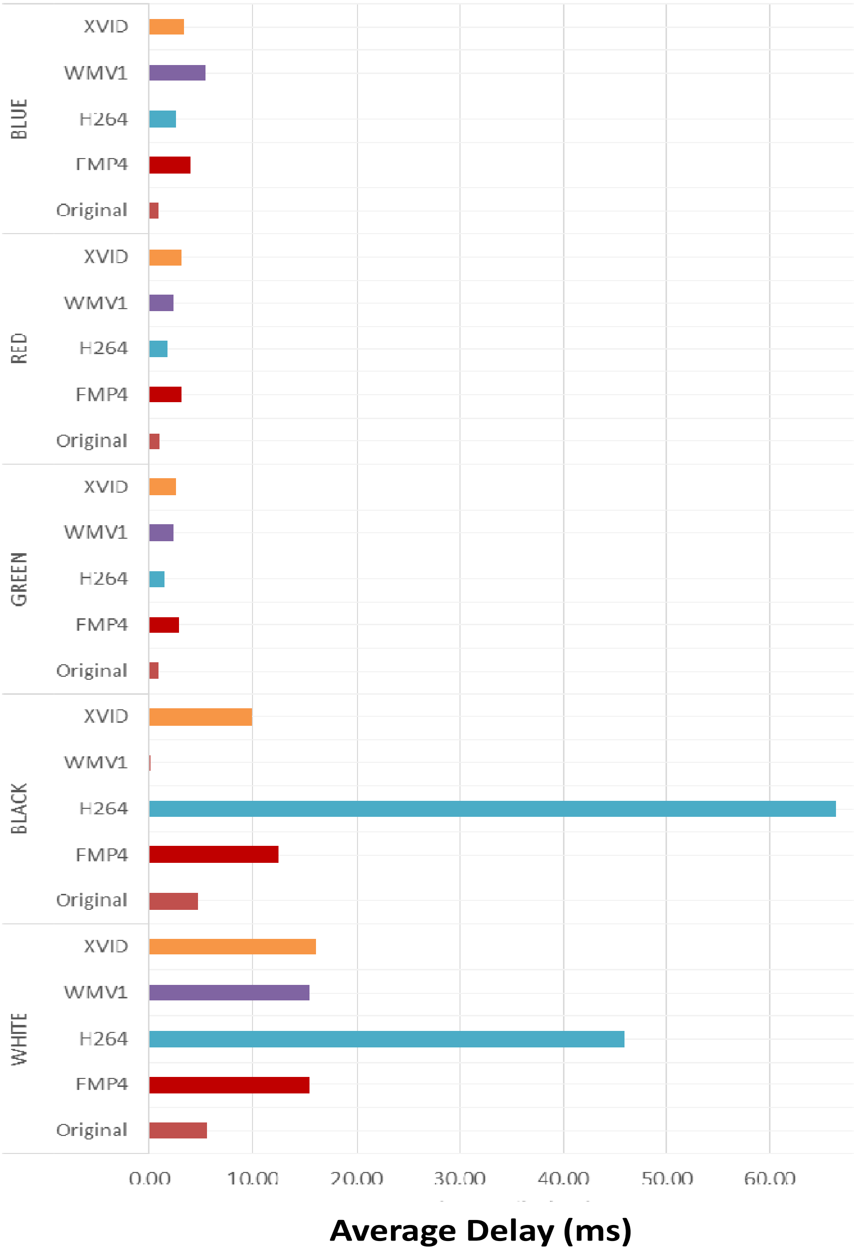


**Figure 5.93.** *Average delay.*

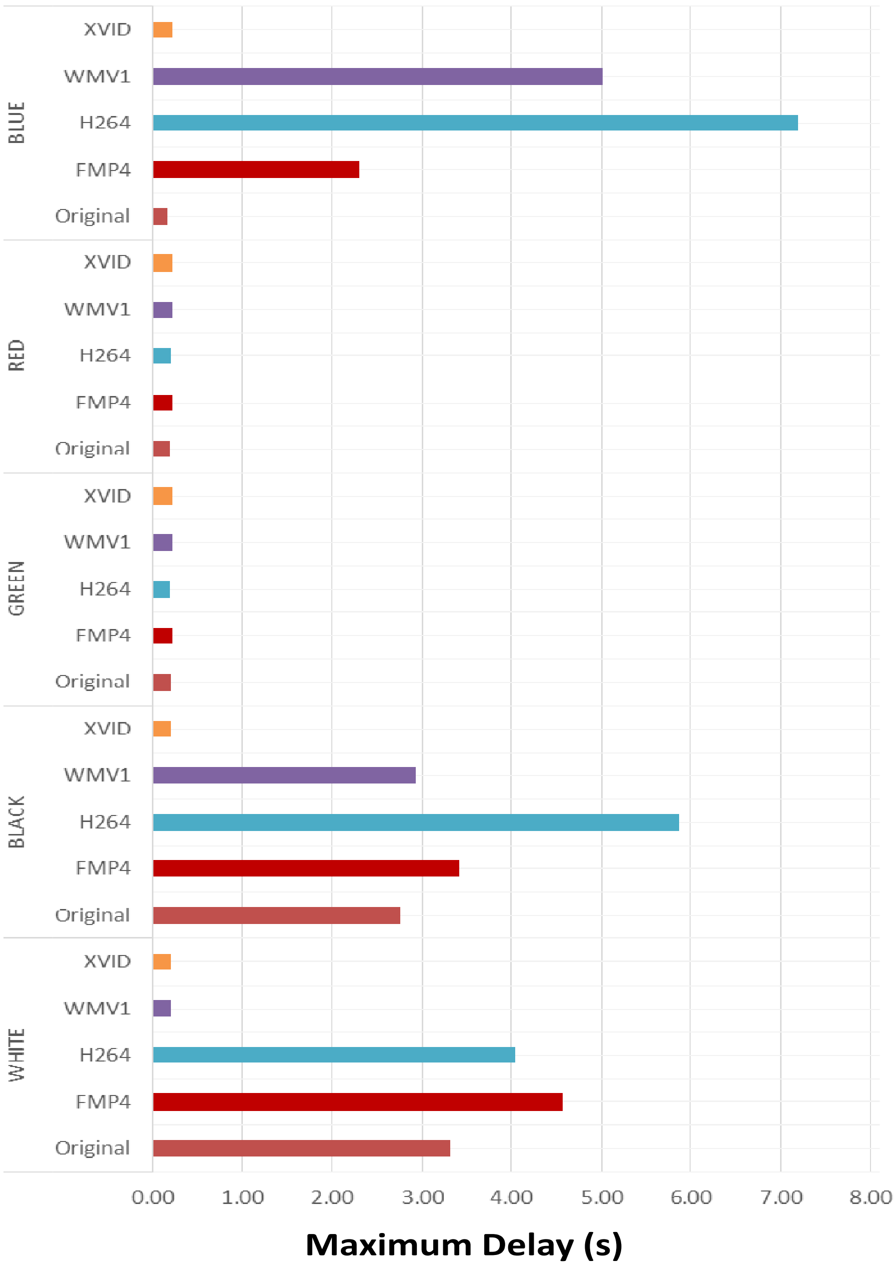


**Figure 5.94.** *Maximum delay.*

#### *5.9.2.5 Results of jitter*

In the following two graphs, Figures 5.95 and 5.96, you can see the comparisons made in terms of the average jitter in the first, see Figure 5.95, and the maximum jitter in the second, see Figure 5.96, obtained for each codec and the maximum delay according to the predominant color.

As can be seen in both graphs, when the color that predominates in the multimedia stream is blue, red or green, the codec that offers the best result in terms of average and maximum jitter is the H264 codec.

In the case of predominant white or black colors, the codec that seems to be the most optimal is WMV1. Since this is a proprietary codec, the XVID codec could be taken as a second option.

These results constitute the seventh and last of the parameters that will be taken into account when making a decision algorithm to establish which compression codec is more suitable according to a predominant color throughout a video.

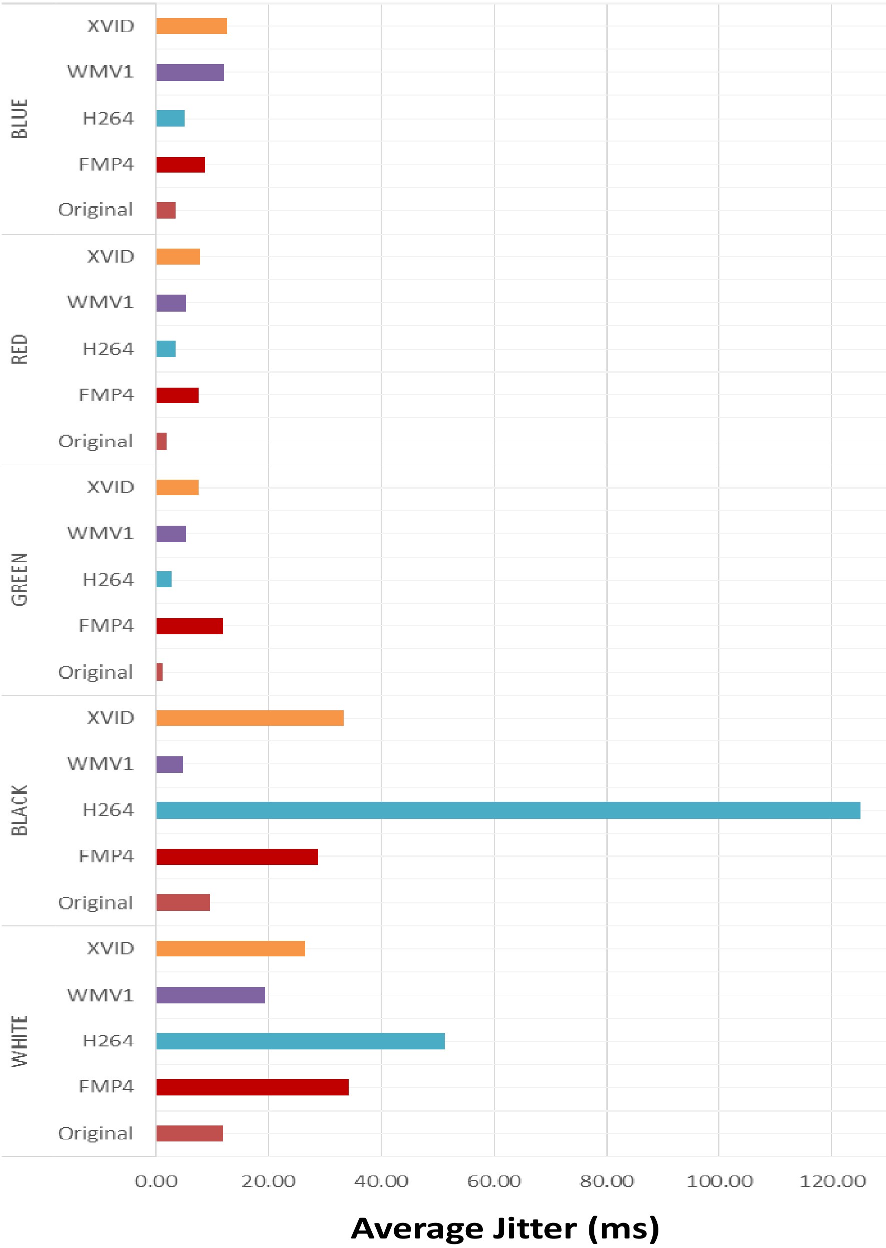


**Figure 5.95.** *Average jitter.*

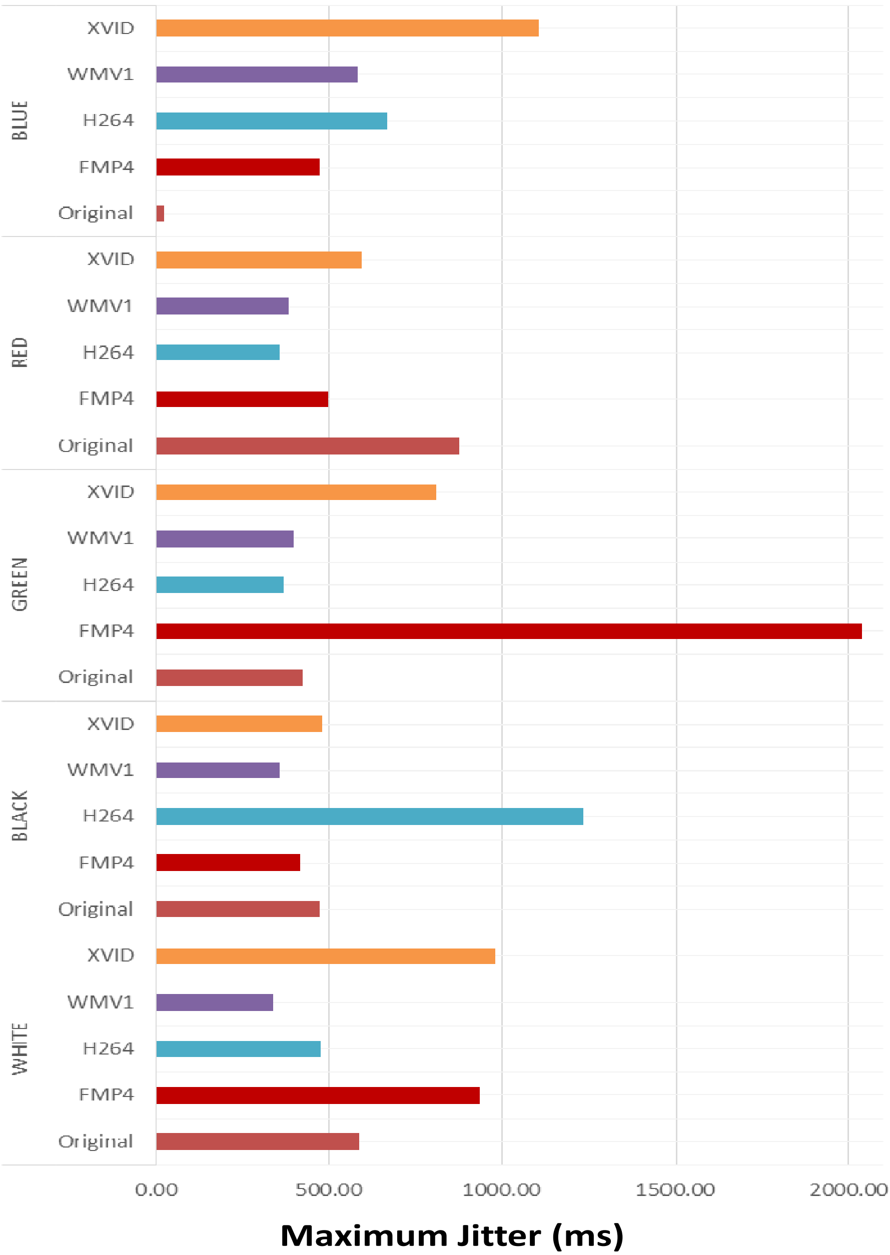


**Figure 5.96.** *Maximum jitter*.

#### 5.9.2.6 Results of packet loss

The data obtained with respect to said study are presented in the Figure 5.97.

In the case of the transmission of a video file whose predominant colors are red, green or blue, the codec with which the packet loss suffered is less and which, therefore, offers a better compression option, is the XVID codec.

In the case that the predominant colors are white or black in the video, it is the codec H264 that offers a better option as a compression codec.

These results constitute the sixth of the parameters that will be taken into account when making a decision algorithm to establish which compression codec is more suitable according to a predominant color throughout a video.

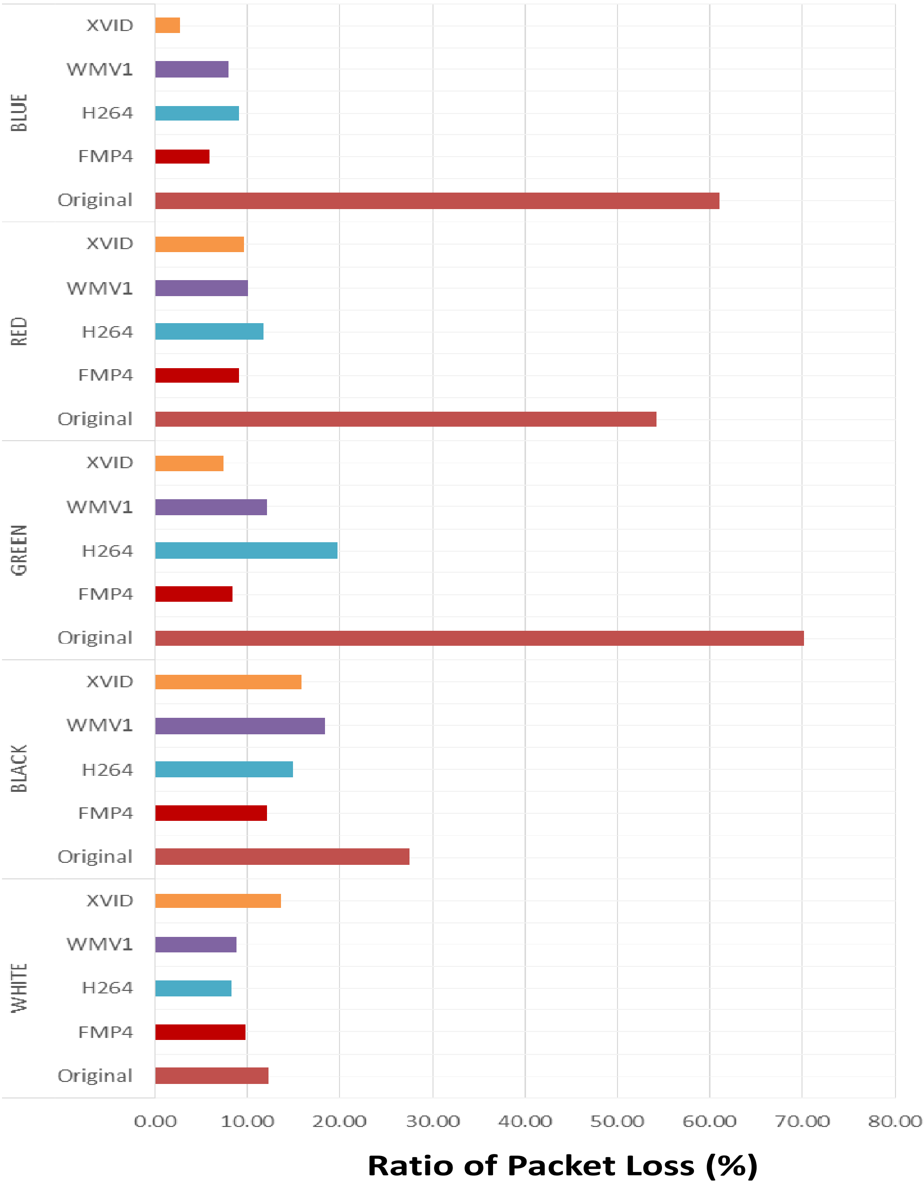


**Figure 5.97.** *Percentage of packet loss.*

#### *5.9.2.7 Subjective perception*

To conclude with the resulting study of recoding each of the video files with a certain compression codec, each of the videos obtained as a result of carrying out said recoding have been reproduced and carefully observed to determine, in a totally subjective way, the relationship of quality between the original video and the recoded one.

In Table 5.17 in which the specified data are exposed being the quality, in descending order: Very Good, Good, Medium, Low and Very Low. This category of classification of image quality has been established based on the quality observed, being for a category 'Very Good' the quality closest to the original and 'Very low' which has been appreciated more distorted.

**Table 5.17.** *Subjective perception of video results.*

| **Output Codec** | **Subjetive Quality** | | | | |
|---|---|---|---|---|---|
| | ***eleph.avi (Red)*** | ***dogs.avi (Green)*** | ***whales.avi (Blue)*** | ***white.avi (White)*** | ***black.avi (Black)*** |
| **APCN** | Very Good | Very Good | Very Good | Very Good | Very Good |
| **FLV1** | Medium | Medium | Medium | Medium | Medium |
| **WMV1** | Low | Medium | Medium | Medium | Medium |
| **XVID** | Medium | Medium | Medium | Medium | Medium |
| **FMP4** | Low | Medium | Medium | Medium | Medium |
| **H264** | Good | Good | Good | Good | Good |

It can be observed in the previous table that, according to the observations made, the Apple ProRes compression codec (APCN) offers a quality almost equal to that of the original video, although for that it has to sacrifice other parameters as seen so far, since is the one that offers the worst results in terms of percentage reduction in size and compression time.

On the other hand, it can be seen how the codec H264 offers a good quality in relation to the original video file, which means that this codec stands out again over the others.

This result constitutes the third of the parameters that will be taken into account when making a decision algorithm to establish which compression codec is most appropriate according to a predominant color throughout a video.

## 5.10 Performance test in videoconference applications

We have made several performance tests using some of the best known videoconference applications used in business, academic and even personal areas. We have made multiple video conferencing sessions with Adobe Conect, Webex and Skype.

The topology used during the test is shown in Figure 5.98. We have used two PCs, with the following features: Intel Core i7-7700 3.6Ghz, RAM 16GB DDR4 2400MHz, integrated network card 10/100/1000, integrated wireless network card and Windows 10 64 bits OS. The network devices we have used have been: a router for accessing the Internet with a connection at 300 Mbps and two Linksys RE6500-EJ access points that support the 802.11 a, b, g and n standards. We have also used two JIAYU smartphones model JY-S3 with eight-core MT6752 processor at 1.7 Ghz., 2GB of Ram and 16 GB of internal memory and Android 5.1 operating system.

We have had to install in all the equipment, PC's and smartphones, the software to make the videoconference, which can be downloaded from web pages and as APPs provided by the manufacturers (Adobe Conect, Cisco Webex and Skype). We have also installed the software that allows us to capture the traffic sent. In the case of PCs we have installed Whireshark, while in smartphones we have used an APP called tPacketCapture [180]. We must indicate that, to be able to install the capture application and that it works correctly in the smartphones, we had to become root, because if we did not do it, even though we started the application, it did not capture packets.

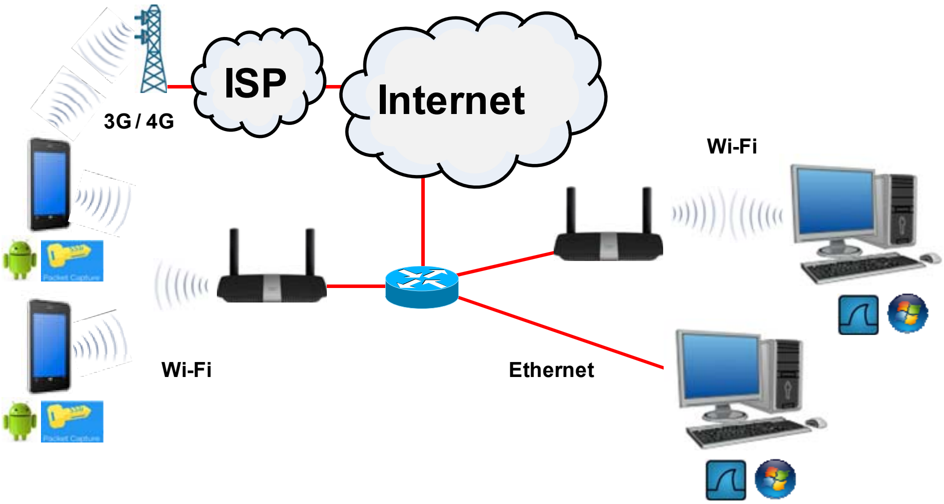


**Figure 5.98.** *Topololgy used in the videoconference performance test.*

To carry out our performance tests, in all cases we have made different captures to observe characteristics of the sent traffic, with each application, and with a duration of 3 minutes.

The data has been captured when the origin of the videoconference was made from different devices, PCs or smartphone (connected by cable or wireless), and the destination was a smartphone that was connected via WiFi, 3G or 4G.

### 5.10.1 Results obtained when using Adobe Connect (WiFi)

In Figure 5.99 it can be seen that when the transmission is made from a PC, both through its Ethernet and wireless interfaces, the traffic increases (approximately 300%) with respect to the transmission made from the smartphone. Due to these results, we consider that Adobe Connect takes into consideration the type of device from which the transmisison is made, PC or smartphone, above the technology used, wired or wireless.

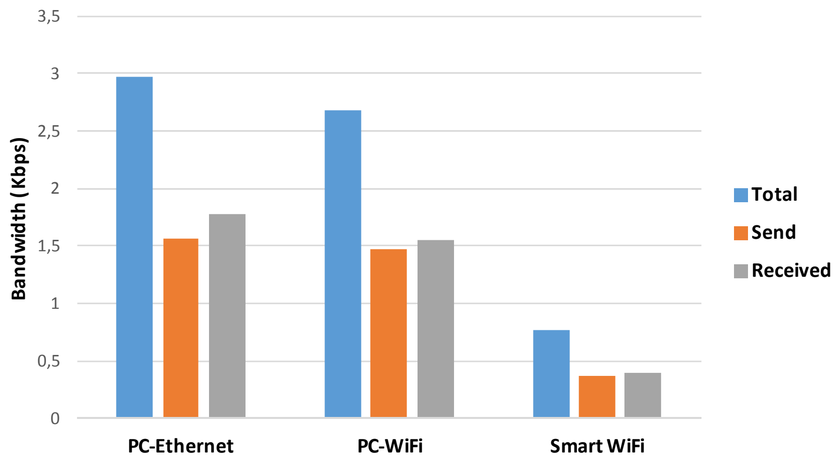


**Figure 5.99.** *Results obtained when using Adobe Connect and Smartphone destination with Wi-Fi.*

### 5.10.2 Results obtained when using Adobe Connect (3G / 4G)

In Figure 5.100, it is observed that the significant differences are more related to the type of device being used in the test, PC or smartphone, regardless to the connection technology (3G / 4G) used in the target device as it happens in Figure 5.99. Unlike when both ends employ WIFI technology, when one uses 3G or 4G the bandwidth consumption is very asymmetric. The bandwidth used by PC multiplies approximately by 4 the bandwidth used by the mobile.

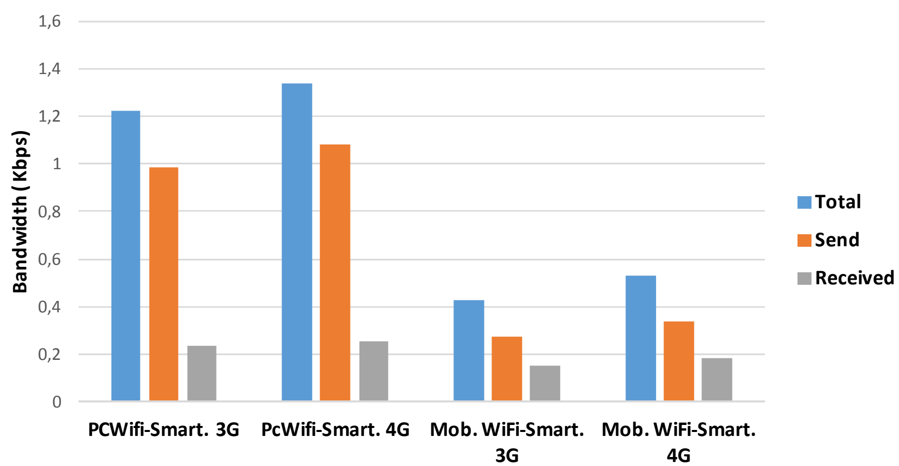

**Figure 5.100.** *Results obtained when using Adobe Connect and Smartphone destination with 3G/4G.*

### 5.10.3 Results obtained when using Cisco Webex (WiFi)

Figure 5.101 shows the bandwidth consumption when the target device is a smartphone connected via WiFi. In general, there are no significant differences when changing the device or technology. It is observed, in the experimental data presented, that when we use WiFi technology, we have obtained lower bandwidth consumption results than with Ethernet.

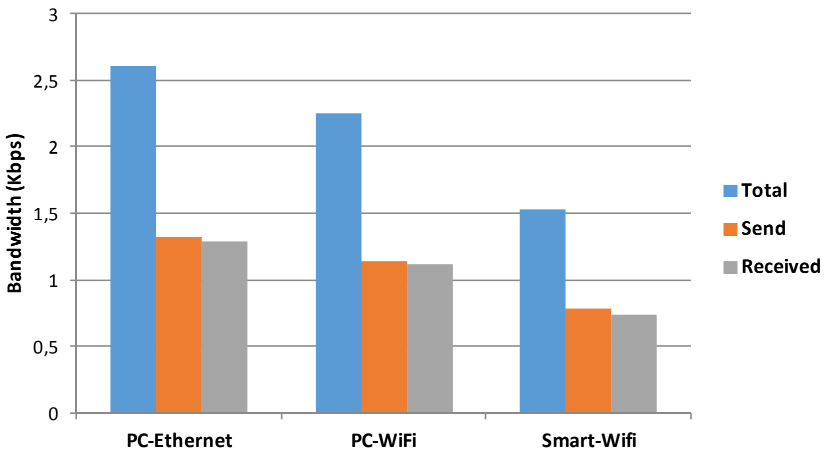


**Figure 5.101.** *Results obtained when using Cisco WebEx and Smartphone destination with Wi-Fi.*

### 5.10.4 Results obtained when using Cisco Webex (3G/4G)

Figure 5.102 shows the results when the target device is connected using 3G / 4G technologies. As can be seen in Figure 5.102, it does not show great differences. The consumption of bandwidth it is slightly asymmetric when the transmission is made from a mobile phone connected by WiFi to a mobile phone connected by 3G or 4G, the smartphone connected via WiFi tends to consume less.

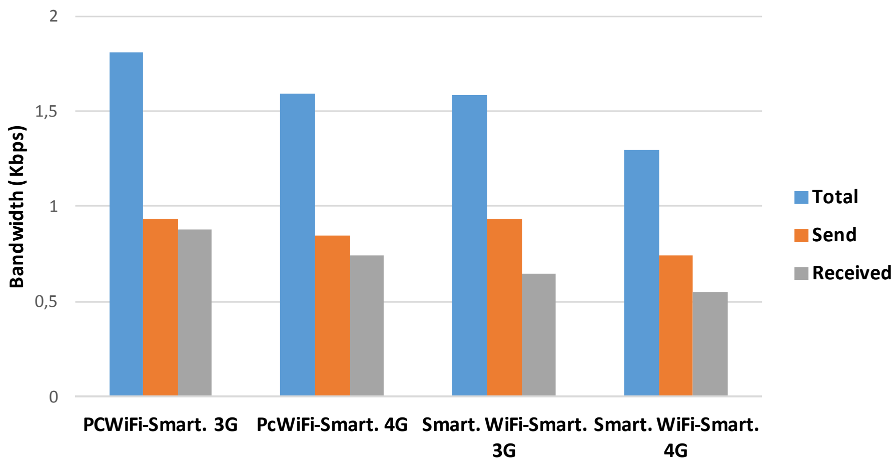


**Figure 5.102.** *Results obtained when using Cisco WebEx and Smartphone destination with 3G/4G.*

### 5.10.5 Results obtained when using Skype (WiFi)

Figure 5.103 shows the results obtained when we have used the Skype application. The results are very similar to those obtained when we use Adobe for the transmission. Significant differences can be observed using PC or smartphone. In the case of establishing the videoconference between a PC and a smartphone or between a smartphone and another smartphone, the transmission is very asymmetric. When the transmission is made between two smartphones the bandwidth saving is very appreciable, reaching a reduction of 80%.

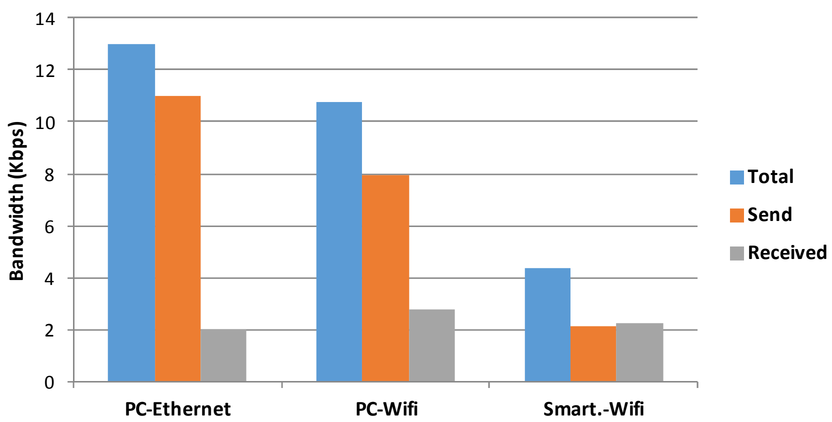

**Figure 5.103.** *Results obtained when using Skype and Smartphone destination with Wi-Fi.*

### 5.10.6 Results obtained when using Skype (3G/4G)

In Figure 5.104 it can be seen that bandwidth consumption is greater when PC is used than when we use smartphones. It is also observed that bandwidth consumption when using 4G technology is slightly higher than when using 3G.

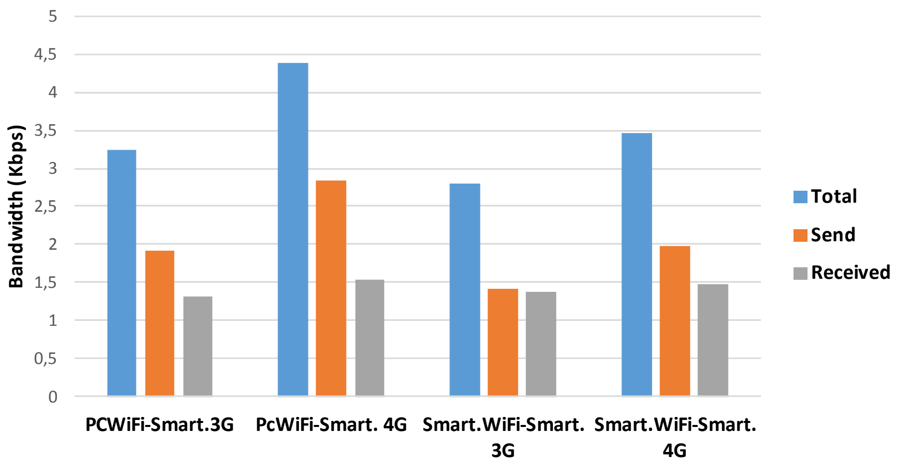


**Figure 5.104.** *Results obtained when using Skype and Smartphone destination with 3G/4G.*

### 5.10.7 Comparative of Cisco Webex vs Adobe Connect in terms of sent packets

In Figure 5.105 we can observe the number of transmitted packets in both Cisco Webex and Adobe Connect. It can be seen a correlation between the bandwidth consumption and the number of transmitted packets, in each of the experimental assumptions. In case of using the fragmentation of packets, depending on the technology or on the type of device, several differences would be appreciated, which may not exist according to the experimental data. A possible improvement could be achieved by modifying the fragmentation of the packets depending on the technologies or the devices.

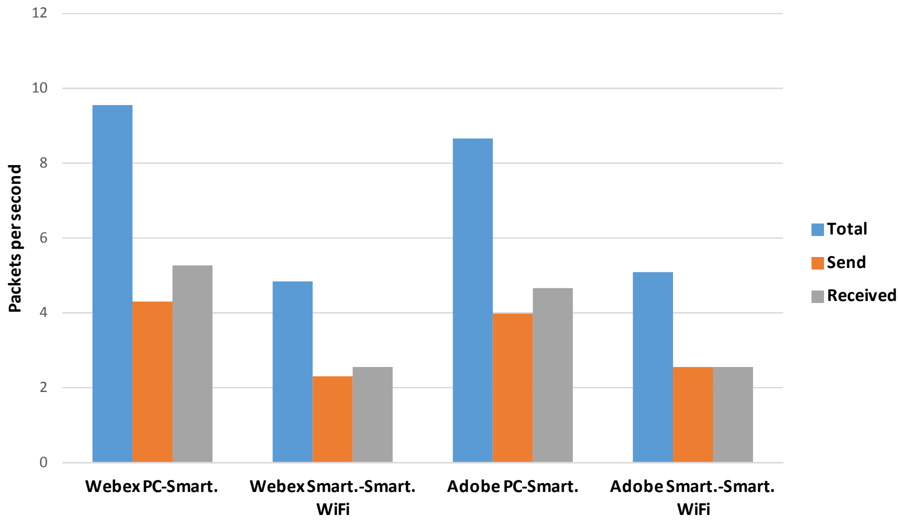


**Figure 5.105.** *Comparative of Cisco WebEx vs. Adobe Connect in terms of sent packets.*

## 5.11 Performance Test of the developed application

In this section, the tests performed with the developed application, are be presented. We have developed a basic application for videoconferencing which implements the features and characteristics of the algorthim and protocol that has been defined and explained in Chapter 4. This application has allowed us to show the validity of our proposal in this thesis with respect to the rest of studied commercial applications in previous section. The main goal is to demonstrate we have improved the QoE of the videoconference users related to other solutions.

The comparative has been performed on three diferent scenarios:

- **Scenario 1.** It focuses on the analysis of the resources of the local device, such as CPU and RAM combined with the Smartphone characteristics (resolution available, camera features, etc.). Following this information, the implementation of our algorithm adapts the videoconference transmission to guarantee the best conditions for the user.

- **Scenario 2.** It focuses on the analysis of the network status from the point of view of the local device. QoS parameters (loss packet, delay, jitter, and bandwidth availability) are observed. When changes in the network conditions happen, the algorithm acts to achieve the better possible QoE.

- **Scenario 3.** Finally, in this last scenario the whole network is anlayzed through the capabilities of SDN. The developed protocol links the movile device with a network managed by SDN, in order to optimize the path of the video transmission used in the videoconference and to minimize the end-to-end delay, jitter and packet loss.

In the next subsections the measurements from our developed application will be referenced as PROTOTYPE.

### 5.11.1 Scenario 1

The experimental set used in this scenario is the same as the one presented in figure 5.98, which was described in section 5.10. Each test has been repeated 10 times and the average has been calculed. The obtained values are presented in figure 5.106. In figure 5.106, we can see how the developed prototype has been able to adapt the bandwidth when the available resources of CPU and RAM have changed.

In order to perform the measurements in this scenario, in addition to the protoype videoconference application, another experimental app has been developed. The only goal of this last application is to spend resources of the device. The application run an infinite loop making some random math calculus and can be adjusted to manage the amount of the resources of the device consumed.

The first column of each application show the bandwidth used for the videoconference when threre are no other applications running in the same device. For the second and third experimental conditions, shown at the second and third columns, the resources consumed by the application described above have been 40 % and 80 % respectively.

From the results, we can see how our prototype application gets worst results when the resources of the device are free (0%). But, when the resources decreases, the algorithm used for the prototype is able to adapt in order to reduce the bandwidth consumption, while the commercial solutions show similar results in the three experimental conditions.

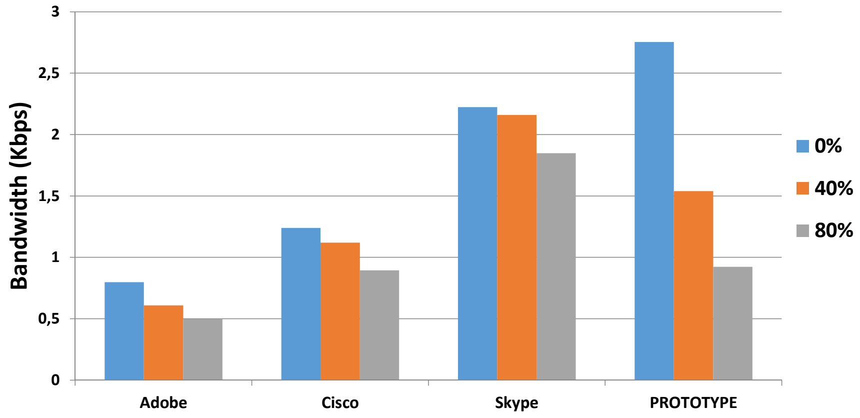


**Figure 5.106** *Results Scenario 1*

### 5.11.2 Scenario 2

In this scenario, we use the same topology as in scenario 1, which was presented in Figure 5.98. As in the previous scenario, we have repeated the tests 10 times for this scenario. Now, our goal is to observe the behavior of our application when the local network parameters (loss packet, delay, jitter, and bandwidth availability) change. Basically we increase the traffic that is sent to the network.

In order to increase network traffic, to achieve congestion, we have developed an application that generates traffic. In addition, the application, which runs on both ends of the network, allows measuring the latency, based on the exchange of standard ICMP packets, between the final devices that perform the videoconference.

As can be seen in Figure 5.107, commercial applications worsen the latency when congestion appears in the network, because they do not make any type of adaptation in the new situation generated, while when using PROTOTYPE, it adapts and maintains low levels of latency.

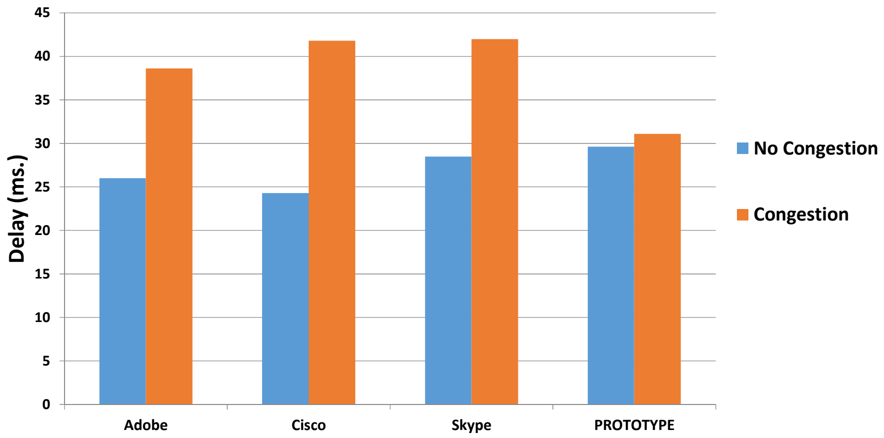


**Figure 5.107** *Results Scenario2.*

### 5.11.3 Scenario 3

Figure 5.108 shows the topology used to perform the tests in scenario 3. In this scenario, we have replaced the router used in previous scenarios by an SDN network.

The SDN network is made up of different devices, among which we have included an SDN controller and several Layer 3 Switches model HP ProCurve 3500YL-24G-PWR Intelligent Edge. These switches support the Open Flow protocol and allow us to work with SDN.

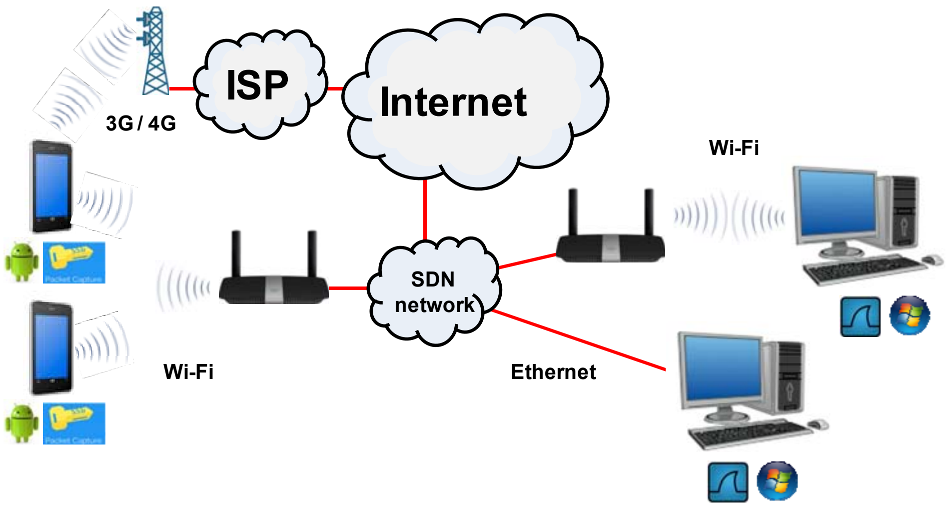

**Figure 5.108.** *Topology used in Scenario 3.*

Our objective, in this scenario, is to observe the correct functioning of our proposal when there is congestion in the transport network. As can be seen in Figure 5.109, what we have created in the SDN network are two basic routes. One of the routes uses a path that crosses a congested network, and the other path avoids the congestion (Congested Path and Not Congested path).

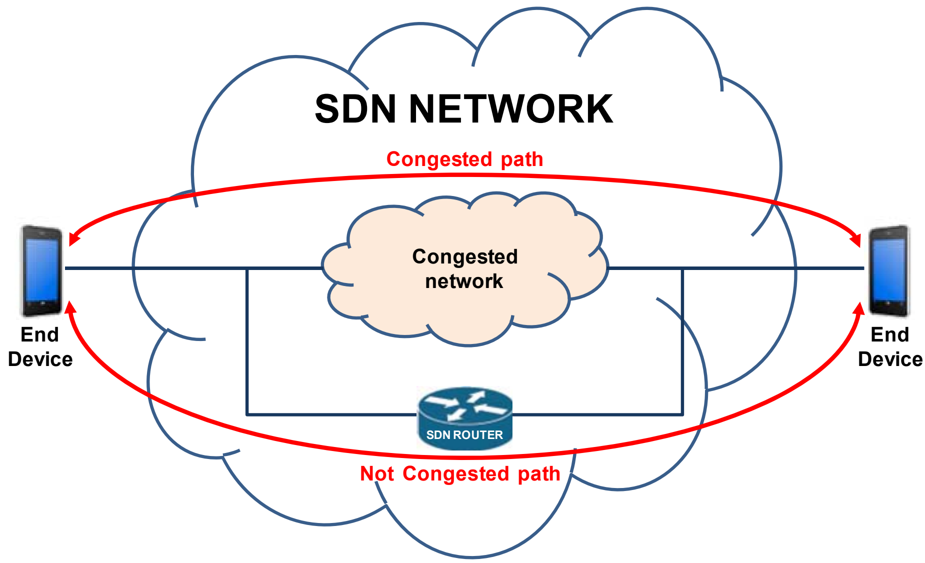


**Figure 5.109.** *SDN Network.*

The SDN controller has been programmed to communicate mobile devices with the SDN network. For this purpose an extension of the OpenFlow protocol has been developed. From PROTOTYPE, which we have installed on mobile devices, we launched an SDN activation request to the controller. From that moment, the SDN controller enters into action so that the traffic is transmitted through the non-congested way.

As can be seen in Figure 5.110, commercial solutions do not change their behavior independently of using or not using the capabilities of SDN technology, since they do not support it.

However, when using PROTOTYPE together with SDN it can be clearly seen how the latency decreases in a very significant way. The

SDN controller sends the traffic through an alternative path, completely free of congestion.

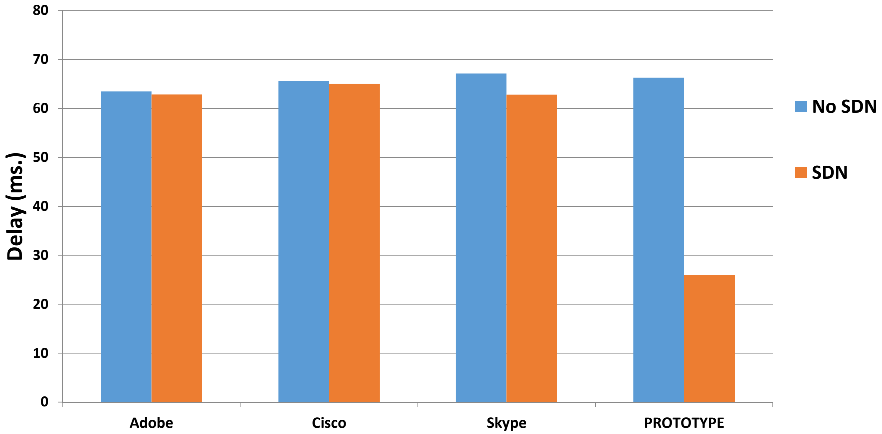


**Figure 5.110.** *Results Scenario 3*

# Chapter 6. Conclusions and future work

## 6.1 Introduction

In this chapter we will present the conclusions obtained at the end of our work. We will also present the research lines that we wish to complete in near future. In addition, we will show a list with the published conference and journal papers derived from this work.

## 6.2 Conclusion

We consider that the previously established objectives of this PhD thesis were reached.

The aim of this work was to allow network operators, application developers, service / content providers and end users to develop, implement and use collaborative services. It is especially useful for those service providers that offer bi-directional and multiparticipatory video connectivity in real time in any mobile browser, application, game, device or service platform.

We have implemented a test bench to observe both the network traffic QoS parameters variations of the received video and the impact in the image quality due to the network parameters variations. We have designed, assembled and configured a WAN network for multimedia traffic transmission. To carry out the test we have used routers, switches, PCs and an access point. We have obtained measurements from the test bench and we have analyzed the significant differences in order to evaluate the QoS parameters basing it on a variance statistical analysis of the different study cases.

We have noticed a significant packet loss when the WAN point-to-point links transmission speed values decrease to near the video transmission average bandwidth (2 Mbps). Some significant changes (but less) are also noticed in the jitter. Delay parameter is not significantly affected.

The study results show that service operators should have both video and QoS tests, in order to have a data transmission statistical database (video, voice, etc.). As far as multimedia communication services demand tends to grow every day, it is necessary to deliver reliability and confidence to customers in order to offer them a good QoS. Therefore, that testbech should be able to reproduce the conditions in both wired and wireless environments.

After performing all these tests, and studied the results, we can say that the main problem of an operator providing IPTV service occurs because it may not have the required bandwidth for all its customers. This can be changed because not all devices receiving IPTV service have the same video quality requirements. We propose a new algorithm where we assume that operators manage their network and can properly control the parameters for the video transmission.

A key step to ensure a satisfactory QoE at the end user side is to deliver the video with enough quality according to the device characteristics. More quality is unnecessary, while it wastes bandwidth, while less quality decreases the QoE. So we propose to have the information coded (or transcoded) according to the device and customer network access characteristics that will be served in different multicast channels.

We have proposed a communication protocol and an algorithm that allows the operator to ensure a good IPTV service to its customers by adjusting the quality of the video according to the used device and the bandwidth hired by the customer. These reasons will provide the highest QoE to the customer.

We have studied different video encodings depending on the bitrate and the encoding time, for Heterogeneous Networks using HTML5. This study allowed us to know the characteristics of each video format. Then, we used this information to design an algorithm, which runs in a server, in order to send the appropriate video type according to the studied characteristics.

In terms of compressed file size, values obtained for PAL are slightly larger than for NTSC system. It happens clearly due to the difference of the frame rate between systems. Therefore, the size values are somewhat higher in the PAL system for both MP4 (H.264) and Ogg (Theora), while in WebM (VP8) are identical in PAL and NTSC.

The most favorable video codec, when taking into account the video quality, the file size and the compressing time, is WebM format, because its final size is the lowest, with the lowest encoding time and with high quality close to the highest bitrates of MP4. We have also shown the graphs corresponding to the compression values obtained from our tests. We have proposed an algorithm that takes into account the browser type, the compression features, the frame rate of the user's region and the needed video quality.

We have studied the performance of different cases in both real topology and virtual topology, when multimedia is being delivered through the topology. The studied parameters have been the consumed bandwidth (throughput), delay and jitter. The study shows how realistic is the emulator Mininet when the performance of multimedia streaming is measured.

We have obtained different minimum values in direct connection than in the connection through switches (using 10Mbps and 100Mbps) in the virtual topology. So, the fact of introducing network devices in the virtual topology affects to the consumed bandwidth, while the real topology does not affect to them. To change from 10Mbps to 100Mbps does not affect to the data in both cases. When we use routers with a WAN link between them at 2 Mb, 4 MB and 8 MB, we obtained the same results in real topology than in the virtual topology. In the delay study the real topology has lower minimum and higher maximum than the virtual topology in all cases. The mean and the maximum peak values of the jitter in the real topology are higher than in the virtual topology when the throughput is high, but when the throughput is low, the maximum peak is higher in the virtual topology.

We have proposed a new smart decision algorithm for selecting the best codec for video streams over IP networks. Bearing in mind the QoS considerations provided by the network and QoE of end users, the algorithm takes the information about the color spectrum to select which video codec should be used to save bandwidth. To develop our algorithm and communication protocol we have performed two tests. The first one analyzes prepared videos with plain background of different colors (black, white, red, blue and green). The second test has analyzed real videos with predominant colors in order to check the quality of image and compression time. As we have seen, different colors suffer different compression level, and the time spent in this process also depends on the dominant color of the video. The dynamic adaptation of the codec used in the video transmission of the same video stream depending on color changes can be translated into a significant network bandwidth saving and improving the QoE perceived by end users. We have also proposed a control protocol for synchronizing the sender and the receiver when this video delivery is going to take place.

We have studied what happens in Videosurveillance when we compress the video with DivX, XviD and H.264 codecs. The video was captured at different hour of the day, so there were videos with different luminance. We have compared these sequences of images based on the compression factor, the time consumed to compress a second of video, % of processor time and the number of inputs and outputs that are in the processor when performing the compression. The comparison has been made taking into account different bit rates for each of the codecs. It has been shown that the choice of any of the three analyzed codecs depends on the limitations or characteristics of the implemented system. If the limitation of the system is the bandwidth of the network, the best one is XviD. If short compression times are needed, the best ones are DivX and XviD. In case the capture device is the limitation of the system, because it has more processes in operation or because the device must capture from several cameras, the best one is H.264.

The use of video surveillance systems in the tasks of environmental monitoring is nowadays widely used. When video flows are delivered through a network, we should try to receive the best video quality. For this reason, we presented the implementation and test of an autonomous video compression system that adapts the video format to the network constraints. The system is based on an algorithm determines which is the best video compression codec for transcoding the video as a function of the measured network parameters and the predominant color of the delivered video. The results have demonstrated that in terms of QoS and QoE, the H264 codec is a good option when the predominant color of videos are black or white, while XVID offer interesting results when red, green or blue are the predominant colors of the video.

We have made several performance tests using some of the best known videoconferencing applications used in business, academic and even personal areas. We have demonstrated the advantages of using the algorithm and protocol proposed and developed in this thesis. In different scenarios, especially in those where there is a scarcity of resources in the devices or the network presents congestion problems, the quality of service is significantly improved and therefore the user's final QoE its improved. Although the bandwidth consumption of the developed prototype is sometimes greater than that of well-established commercial solutions with a long history, the adaptability that our prototype presents makes it much more versatile and flexible when it comes to impelling it in both a wide range of mobile devices (with very different hardware and software characteristics) as well as on network platforms with different characteristics.

## 6.3 Future Research

In future works, we intend to use the test bench defined to test the received video quality using different types of queues, video streaming applications, and different video codecs performance and the received video quality. Moreover, due to the proliferation of the video streaming applications for Content Delivery Networks (CDNs), Peer-to-Peer (P2P) networks, Multimedia Cloud Computing, etc., we will use the implemented test bench in new streaming applications. In addition, it will allow us to test on any device (laptops, phones, etc.) the performance and quality of different video and audio codecs after the delivery.

Among the research lines that we would like to include in the designed protocol and algorithm, we find: further test of the characteristics of IPTV set-top boxes, study more video codecs, find the exact thresholds where video is received with high quality.

We will investigate the influence of secondary colors in a sequence with a particular predominant color or use group of colors. We will also investigate the frequency with which the execution of the algorithm must be performed to achieve the best balance between improvements of videos and the system overhead due to the repeated execution of process. Moreover, we will investigate the influence of secondary colors in a sequence with a certain predominant color or group of colors. Furthermore, as well as other authors have done for VoIP, we would like to improve the code implemented in Python and add more functionality in terms of codec selection, video conversion formats and the selection of type of device among others. It could be interesting to add other parameters related to image properties such as luminance or exposure.

Finally, we have also in mind to investigate the frequency with which, the execution of the algorithm proposed in Chapter 4 must be executed to achieve the best balance between the improvement of the received video and the overload of the system, because the final goal is to have an adaptive video streaming system while maintaining the QoE of the final user.

## 6.4 Publications derived from the PhD Thesis

Next papers are derived (or very related) from the research work presented in this dissertation. The papers that are directly related with the dissertation are the following ones.

**Journal papers**:

Albert Rego, Alex Canovas, **Jose M. Jimenez**, Jaime Lloret, An Intelligent System for Video Surveillance in IoT Environments, IEEE Access, 18 June 2018, PP. 1-1, (Q1, IF = 3,557)

**Jose M. Jimenez**, Juan R. Diaz, Jaime Lloret, Oscar Romero, MHCP: Multimedia Hybrid Cloud Computing Protocol and Architecture for Mobile Devices, IEEE Network (Q1, IF = 7,197, In press)

Albert Rego, Sandra Sendra, **Jose M. Jimenez**, Jaime Lloret, Dynamic Metric OSPF-Based Routing Protocol for Software Defined Networks, Cluster Computing (Q2, IF = 1,601, In press)

Alejandro Canovas, **Jose M. Jimenez**, Oscar Romero, Jaime Lloret, Multimedia Data Flow Traffic Classification using Intelligent Models based on Traffic Patterns, IEEE Network (Q1, IF = 7,197, In press)

**Jose M. Jimenez**, Oscar Romero, Jaime Lloret, Juan R. Diaz, Energy Savings Consumption on Public Wireless Networks by SDN Management, Mobile Networks and Application, 30 November 2016, pp. 1 – 11 (Q1, IF = 3,259)

Miran Taha, **Jose M. Jimenez**, Alejandro Canovas, Jaime Lloret, Intelligent Algorithm for Enhancing MPEG-DASH QoE in eMBMS, Network Protocols and Algorithms, Vol. 9, Issue 3-4 (2017), pp. 94-114

Irene Mateos-Cañas, Sandra Sendra, Jaime Lloret, **Jose M. Jimenez**, Autonomous video compression system for environmental monitoring, Network Protocols and Algorithms, Vol. 9, No. 1-2 (2017), pp. 48 – 70

**Jose M. Jimenez**, Miran Taha, Sandra Sendra, Jaime Lloret, Interactive Videos in IPTV using Hypervideolinks, Network Protocols and Algorithms, Vol. 9, Issue 3-4 (2017), pp. 77-93

**Jose M. Jimenez**, Oscar Romero, Albert Rego, Jaime Lloret, Analyzing the Performance of Software Defined Networks vs Real Networks, International Journal on Advances in Networks and Services, vol 9 no 3 & 4, 2016, pp 107 – 116

**Jose M. Jimenez**, Oscar Romero, Albert Rego, Avinash Dilendra, Jaime Lloret, Study of Multimedia Delivery over Software Defined Networks, Network Protocols and Algorithms, Vol 7, No 4 (2015)

**Conference papers**:

Jaime Lloret, Jesus Tomas, Oscar Romero, **Jose M. Jimenez**, Albert Rego, Belen Carro, Antonio Sánchez-Esguevillas, Manuel López-Martín, Santiago Egea, Multimedia Services Distribution Using Adaptive and Cognitive SDNs, XIII Jornadas de Ingenieria Telemática (JITEL 2017), 27-29 Septiembre 2017, Valencia (España)

Laura Garcia, **Jose M. Jimenez**, Miran Taha, Jaime Lloret, Video artifact evaluation based on QoS and objective QoE parameters, 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI 2017), 13-16 Sept. 2017, Udupi, India.

Miran Taha, Laura Garcia, **Jose M Jimenez**, Jaime Lloret, SDN-based throughput allocation in wireless networks for heterogeneous adaptive video streaming applications, 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 26-30 June 2017, Valencia, Spain (CORE B)

Sandra Sendra, Albert Rego, Jaime Lloret, **Jose M. Jimenez**, Oscar Romero, Including artificial intelligence in a routing protocol using Software Defined Networks, 2017 IEEE International Conference on Communications (ICC), 21-25 May 2017, Paris, France (CORE B)

**Jose M. Jimenez**, Jaime Lloret, Juan R. Diaz, Raquel Lacuesta, Test Bench to Test Protocols and Algorithms for Multimedia Delivery, 11th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2016), Hangzhou, China, June 14-15, 2016. pp 124-134

**Jose M Jimenez**, Oscar Romero, Albert Rego, Avinash Dilendra, Jaime Lloret, Performance study of a software defined network emulator, (2016) The twelfth advanced international conference on telecommunications (AICT 2016), May 22–26, 2016, Valencia, Spain

**Jose M. Jimenez**, Alejandro Canovas, Juan R. Diaz, Jaime Lloret, Estudio de QoS y QoE y Propuestas de Sistemas para la Mejora de los Servicios Multimedia e IPTV, XII Jornadas de Ingenieria Telemática (JITEL 2015), 14 al 16 de octubre de 2015, Palma de Mallorca (España)

**Jose M. Jimenez**, Alejandro Canovas, Andrés López, Jaime Lloret, A New Algorithm to Improve the QoE of IPTV Service Customers, 2015 IEEE International Conference on Communications (ICC), 8-12 June 2015, London, UK, (CORE B)

Andrés López-Herreros, Alejandro Canovas, **Jose M. Jimenez**, Jaime Lloret, A New IP Video Delivery System for Heterogeneous Networks using HTML5, 2015 IEEE International Conference on Communications (ICC), 8-12 June 2015, London, UK, (CORE B)

**Jose M. Jimenez**, Juan R. Diaz, Sandra Sendra, Jaime Lloret, Choosing the Best Video Compression Codec Depending on the Recorded Environment, 2014 IEEE Global Communications Conference (GLOBECOM), 8-12 Dec. 2014, Austin, TX, USA. (CORE B)

Jaime Lloret, Pedro V. Mauri, **Jose M. Jimenez**, Juan R. Diaz, 802.11g WLANs Design for Rural Environments Video-surveillance, ICDT '06. International Conference on Digital Telecommunications, 29-31 Aug. 2006, Cote d'Azur, France

Ruth Herrero Doñate, Jaime Lloret, **Jose M. Jimenez**, Estudio de códecs de compresión MPEG-4 para su aplicación a la videovigilancia, XX Simposium Nacional de la Unión Científica Internacional de Radio (URSI2005), 14-16 de Septiembre de 2005,Gandia (Valencia)

Papers under review or accepted with major or minor revision:

**Jose M. Jimenez**, Oscar Romero, Jaime Lloret, Juan R. Diaz, A new Architecture based on SDN for Energy Savings in network devices to interconnect Data Centers, IEEE Communications Magazine (Q1, In review)

During last years I have collaborated in the development of several models, proposals and experimental set-ups which have served as first approach for the final purpose of this dissertation. They have helped me to build the foundations on which the final architecture is grounded. These preliminary works have not only contributed substantially to the progress of this research, but also opened new lines of related research that have subsequently been or will be developed by other members of the research group with whom I have been collaborating. The published papers corresponding to this work are presented in a prominent place within this thesis.

Laura Garcia, **Jose M. Jimenez**, Miran Taha, Jaime Lloret, Wireless Technologies for IoT in Smart Cities, Network Protocols and Algorithms, Vol. 10, Issue 1 (2018), pp. 23-64

Carlos Barambones, Laura García, **Jose M. Jiménez**, Jaime Lloret, A New Tool to Test the IP Network Performance, Network Protocols and Algorithms, Vol 8, No 2 (2016)

Alain Chas, **Jose M. Jimenez**, Andrés López-Herreros, Alejandro Canovas, Jaime Lloret, AC TV Manager: Nuevo sistema de gestión para IPTV basado en software libre, JITEL2015, 14 – 16 de octubre de 2015, Palma de Mallorca, España

Rubén Tortosa, **Jose M. Jiménez**, Juan R. Diaz, Jaime Lloret, Optimal Codec Selection Algorithm for Audio Streaming, IEEE Global Communications Conference Workshops (IEEE Globecom 2014), Austin, Texas (USA), December 8-12, 2014.

There are other published papers which have some relation to the work presented in this Ph.D.:

Sendra S , Lloret J , Lacuesta R , **Jimenez JM**, Energy Efficiency in Cooperative Wireless Sensor Networks, Mobile Networks and Applications, 30 November 2016, 1-10 (2016) (Q1, IF = 3,259)

Sandra Sendra, Laura Garcia, **Jose M Jimenez**, Jaime Lloret, Low-Cost Vehicle Driver Assistance System for Fatigue and Distraction Detection, International Conference on Future Intelligent Vehicular Technologies, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol 185, 2016, pp 69 – 78

**Jose M. Jimenez**, Jaime Lloret, Juan R. Diaz, Test Bench to Test Protocols and Algorithms for Multimedia Delivery, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol 177, 2016, pp 124 – 134

Sandra Sendra, Jaime Lloret, Raquel Lacuesta, **Jose M. Jimenez**, Cooperative Monitoring of the Delivery of Fresh Products, Lecture Notes in Computer Science, Vol. 9320, 2015, pp. 76 – 86

Diaz, Juan R., Lloret, J., **Jimenez, Jose M.**, Sendra, MWAHCA: a multimedia wireless Ad Hoc cluster architecture, The Scientific World Journal, 2014 (Q2, IF = 3,591)

Diaz, Juan R., Lloret, J., **Jimenez, Jose M.**, Hammoumi, M., A new multimedia–oriented architecture and protocol for wireless ad hoc networks, International Journal of Ad Hoc and Ubiquitous Computing 16 (1), 14-25, 2014 (Q4, IF = 0,554)

Diaz, Juan R., Lloret, J., **Jimenez, Jose M.**, Sendra, S., Rodrigues, JJPC, Fault tolerant mechanism for multimedia flows in wireless ad hoc networks based on fast switching paths, Mathematical Problems in Engineering, 2014, (Q3, IF = 3,190)

# References

[1] Kandula, S., Sengupta, S., Greenberg, A., Patel, P., & Chaiken, R. The nature of data center traffic: measurements & analysis. In Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. Chicago, Illinois, USA, November 4-6 2009. Pp. 202-208

[2] Astuto B., Nunes A., Mendonca M., Nguyen X.-N., Obraczka K., Turletti T., A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks, IEEE Communications Surveys & Tutorials, Volume: 16, Issue: 3. Pp. 1617 - 1634. Third Quarter 2014. DOI:10.1109/SURV.2014.012214.00180xxxx

[3] Software-Defined Networking (SDN) Definition at: https://www.opennetworking.org/sdn-resources/sdn-definition

[4] Gross T., Sayama H., Adaptative networks, Understanding Complex Systems, DOI 10.1007/978-3-642-01284-6_1, NECSI Cambridge/Massachusetts 2009

[5] A. H. Sayed, "Diffusion adaptation over networks," in E-Reference Signal Processing, R. Chellapa and S. Theodoridis, Eds. Amsterdam, The Netherlands: Elsevier, available online at http://arxiv.org/abs/1205.4220

[6] R.W. Thomas, L.A. DaSilva, A.B. MacKenzie, "Cognitive networks", Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Baltimore, MD, USA, November 8–11, 2005.

[7] A. J. Estepa, R. Estepa, J. Vozmediano, P. Carrillo, Dynamic VoIP codec selection on smartphones, Network Protocols and Algorithms, Vol 6, No 2 (2014). Pp. 22-37

[8] J. Lloret, C. Palau and M. Esteve, Structuring connections between content delivery servers groups, Future Generation Computer Systems. Vol. 24, Issue 3, pp. 191-201 March 2008

[9] L. Fabrega, T. Jove, A review of the architecture of admission control schemes in the Internet, Network Protocols and Algorithms, Vol 5, No 3 (2013). Pp.1-32

[10] Recommendation BT.601. Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. March 2011. Available at: http://www.itu.int/rec/R-REC-BT.601/

[11] J. Lloret, M. Garcia, A. J. Ferrer and M. Morant, HDTV compression analysis for storage and transmission over Internet, The 5th WSEAS

International Conference on DATA NETWORKS, communications and computers (DNCOCO 2006), Bucharest (Romania), 16-18 October 2006

[12] D. Bri, A. Canovas, J. Tomas, J. Lloret, An Intelligent System to Detect the Type of Devices Sending and Receiving Data in the Network, Network Protocols and Algorithms, Vol 5, No 2 (2013). pp. 72-91

[13] The Consumer Barometer is a tool to help you understand how people use the Internet across the world. Available at: https://www.consumerbarometer.com/en/ (last accessed on 14/5/2018)

[14] Videoconferencia. Available at: http://dle.rae.es/?id=bmaBgk4 (last accessed on 14/5/2018)

[15] Cabero J., La videoconferencia como instrumento educativo, Nuevas Tecnologías aplicadas a la educación, Madrid, Síntesis, pp. 97 - 110

[16] Video Conferencing Expected to be Preferred Business Communications Tool in 2016 According to New Survey on Global Video Conferencing Trends and Etiquette. Available at: http://www.polycom.co.uk/company/news/press-release-filter/2013/20131015.html (last accessed on 14/5/2018)

[17] CU-SeeMe Welcome page. Available at: http://ftp.icm.edu.pl/packages/cu-seeme/html/xxxWelcome.html. (last accessed on 14/5/2018)

[18] Dorcey T., CU-SeeMe Desktop Video Conferencing Software, in Connexions 9, 3 (March 1995)

[19] Messenger. Available at: https://www.messenger.com/. (last accessed on 14/5/2018)

[20] FaceTime Be in two places at once. Available at: https://web.archive.org/web/20121205020704/http://www.apple.com/ios/facetime/ (last accessed on 14/5/2018)

[21] Google Duo. Available at: https://duo.google.com/ (last accessed on 14/5/2018)

[22] Say Allo Chat on Android, iOS, and web. Available at: https://allo.google.com/ (last accessed on 14/5/2018)

[23] ICQ. Available at: https://icq.com/windows/en (last accessed on 14/5/2018)

[24] AOL. Available at: https://www.aol.com/ (last accessed on 14/5/2018)

[25] @Mail.Ru. Available at: https://mail.ru/ (last accessed on 14/5/2018)

[26] Jitsi. The most innovative open source video conferencing community. Available at: https://jitsi.org/ (last accessed on 14/5/2018)

[27] Line. Available at: https://line.me/en/ (last accessed on 14/5/2018)

[28] Un Skype para todos tus dispositivos. Available at: https://www.skype.com (last accessed on 14/5/2018)

[29] A New Kind of Instant Messaging. Available at: https://tox.chat/ (last accessed on 14/5/2018)

[30] National Security Agency | Central Security Service Defending our Nation. Securing the Future. Available at: https://www.nsa.gov/ (last accessed on 14/5/2018)

[31] Viber connects over 900 million users. Available at: https://www.viber.com/ (last accessed on 14/5/2018)

[32] Rakuten. Available at: https://global.rakuten.com/corp/ (last accessed on 14/5/2018)

[33] Video Telemedicine That Works. Available at: https://vsee.com/ (last accessed on 14/5/2018)

[34] Simple. Seguro. Mensajería confiable. Available at: https://www.whatsapp.com/ (last accessed on 14/5/2018)

[35] Connecting a billion people with calls, chats, and more. Available at: https://www.wechat.com/en/ (last accessed on 14/5/2018)

[36] The most secure collaboration platform. Available at: https://wire.com/en/ (last accessed on 14/5/2018)

[37] The new yahoo messenger. Available at: https://messenger.yahoo.com/ (last accessed on 14/5/2018)

[38] Powering mission critical web conferencing solutions across devices. Available at: https://www.adobe.com/products/adobeconnect.html (last accessed on 14/5/2018)

[39] Avaya Equinox® Meetings Online. Your High-Performance Meeting Room in the Cloud. Available at: https://www.avaya.com/en/product/avaya-equinox-meetings-online/ (last accessed on 14/5/2018)

[40] About BBM Meetings. Available at: http://help.blackberry.com/en/bbm-meetings-security/latest/bbm-meetings-security/awi1417549079994.html (last accessed on 14/5/2018)

[41] Bliz A new meeting and collaboration tool brought to you by TeamViewer. Available at: https://www.blizz.com/en/teamviewer-meeting-info/ (last accessed on 14/5/2018)

[42] The Meetings Platform for the Modern Workplace. Available at: https://www.bluejeans.com/ (last accessed on 14/5/2018)

[43] Welcome to Camfrog!. Available at: https://kopano.com/ (last accessed on 14/5/2018)

[44] Trust your online meetings to Cisco WebEx. Available at: https://www.cisco.com/c/en/us/products/conferencing/web-conferencing/index.html (last accessed on 14/5/2018)

[45] Cisco Jabber. Available at: https://www.cisco.com/c/en/us/products/unified-communications/jabber/index.html (last accessed on 14/5/2018)

[46] Ekiga. Available at: https://www.ekiga.org/ (last accessed on 14/5/2018)

[47] HD Audio and Video Conferencing - Meet face to face anyplace. Available at: https://www.fuze.com/unified-communications/collaboration#hd-audio (last accessed on 14/5/2018)

[48] The enterprise co browse company. Available at: https://ww2.glance.net/ (last accessed on 14/5/2018)

[49] Habla con tus amigos y familiares. Available at: https://hangouts.google.com/ (last accessed on 14/5/2018)

[50] Face-to-face conversations come as standard with GoToMeeting video conferencing software. Available at: https://www.gotomeeting.com/en-gb/features/hd-video-conferencing (last accessed on 14/5/2018)

[51] Business collaboration simplified. Available at: https://highfive.com/ (last accessed on 14/5/2018)

[52] Your Data, Your Terms. Available at: https://kopano.com/ (last accessed on 14/5/2018)

[53] Extraordinary video conferencing. Available at: https://www.lifesize.com/ (last accessed on 14/5/2018)

[54] Video Conferencing, Web Conferencing & Webinar Solutions. Available at: http://www.megameeting.com/ (last accessed on 14/5/2018)

[55] Please Allow GlobalMeet to Reintroduce Itself. Available at: https://www.pgi.com/ (last accessed on 14/5/2018)

[56] Es momento de replantear la colaboración. Available at: http://www.polycom.es/ (last accessed on 14/5/2018)

[57] Fast, simple, secure. Available at: https://signal.org/ (last accessed on 14/5/2018)

[58] Free Video Conferencing Server and Apps. Available at: https://trueconf.com/ (last accessed on 14/5/2018)

[59] Free and secure calls and messages to anyone, anywhere. Available at: https://www.viber.com/es/ (last accessed on 14/5/2018)

[60] In today's digital world, the human touch matters more than ever. Enrich experiences with the best real-time HD video conferencing solutions. Available at: https://www.vidyo.com/ (last accessed on 14/5/2018)

[61] #1 Video Conferencing and Web Conferencing Service. Available at: https://zoom.us/ (last accessed on 14/5/2018)

[62] Internet Engineering Task Force (IETF). Available at: http://www.ietf.org/ (last accessed on 14/5/2018)

[63] What's Software-Defined Networking (SDN). (Online article). Available at sdx central website: https://www.sdxcentral.com/resources/sdn/what-the-definition-of-software-defined-networking-sdnYoung, G. O., "Synthetic structure of industrial plastics", in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64. (last accessed on 14/5/2018)

[64] Software-Defined Networking (SDN) Definition. (Online article). Available athttps://www.opennetworking.org/sdn-resources/sdn-definition (last accessed on 14/5/2018)

[65] Software-Defined Networking: A Perspective from within a Service Provider Environment. Available at: https://tools.ietf.org/pdf/rfc7149.pdf (last accessed on 14/5/2018)

[66] What is OpenFlow?. Available at: http://archive.openflow.org/wp/learnmore (last accessed on 14/5/2018)

[67] Software-Defined Networking: The New Norm for Networks. Available at Open Network Foundation website: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf (last accessed on 14/5/2018)

[68] NFV standard. Available at IETF website: https://portal.etsi.org/tb.aspx?tbid=789&SubTB=789,795,796,801,800,798,799,797,802 (last accessed on 14/5/2018)

[69] Perrin, S., Hubbard, S., Practical implementation of SDN & NFV in the WAN, White paper, Heavy Reading, October 2013, https://networkbuilders.intel.com/docs/HR-Intel-SDN -WP.pdf

[70] ITU: Committed to connecting the world. Available at: https://www.itu.int/en/Pages/default.aspx (last accessed on 14/5/2018)

[71] ITU: Committed to connecting the world. About ITU Telecom. Available at: https://www.itu.int/en/itutelecom/Pages/default.aspx (last accessed on 14/5/2018)

[72] Integrated services digital network. Available at: https://www.itu.int/rec/T-REC-I/en (last accessed on 14/5/2018)

[73] ITU-T H.323 v7 (12/2009). Available at: https://www.itu.int/itu-t/recommendations/rec.aspx?rec=10638&lang=en (last accessed on 14/5/2018)

[74] Lu X., Tao S., El Zarki M., Guérin R., Quality based Adaptive Video over the Internet. Dept of ESE, University of Pennsylvania, Philadelphia, PA 19104 (2003)

[75] Lee T. K., Chai T. Y., Ngoh L. H., Shao X., Teo J. C. M., Zhou L., An IMS-based testbed for real-time services integration and orchestration. IEEE Asia-Pacific Services Computing Conference, (APSCC 2009). Singapore, December 7-11, 2009

[76] Mu M., Mauthe A., Casson J., Tyson G., Garcia F., LA1 TestBed: evaluation testbed to assess the impact of network impairments on video quality. 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops, (TridentCom 2009). April 6~8, 2009, Washington DC

[77] Kuwadekar A., Al-Begain K., User Centric Quality of Experience Testing for Video-on-Demand (VoD) over IMS. First International Conference on Computational Intelligence, Communication Systems and Networks (CICSYN'09), Indore, India, 23-25 July 2009

[78] Lee, D.; Joo, H.; Song, H., An effective channel control algorithm for integrated IPTV services over DOCSIS CATV networks. IEEE Transactions on Broadcasting, Vol. 53, no 4, 2007, pp. 789-796.

[79] Zapater, M. N., Bressan, G., A proposed approach for quality of experience assurance of IPTV. First International Conference on the Digital Society (ICDS '07), Guadeloupe, 2007, pp. 25-25.

[80] Garcia, M., Canovas, A., Edo, M., Lloret, J., A QoE Management System for Ubiquitous IPTV Devices, Third International Conference on

Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '09), Sliema (Malta), 11-16 Oct. 2009, pp 147-152.

[81] Lloret, J., Garcia, M.; Atenas, M., Canovas,A., A QoE management system to improve the IPTV network, International Journal of Communication Systems, John Wiley & Sons, Ltd., Vol. 24 nº 1, 2011/1/1, pp. 118-138.

[82] Lloret, J., Canovas,A., Tomas, J.; Atenas, M., A network management algorithm and protocol for improving QoE in mobile IPTV, Computer Communications, Vol 35, Issue 15, 1 September 2012, pp. 1855-1870.

[83] Lloret, J., Canovas, A., Rodrigues, J. J., Lin, K. A network algorithm for 3D/2D IPTV distribution using WiMAX and WLAN technologies, Multimedia tools and applications, Vol 67 N. 1, pp. 7-30.

[84] Bramberger M., Brunner J., Rinner B., Schwabach H., Real-time video analysis on an embedded smart camera for traffic surveillance. Real-Time and Embedded Technology and Applications Symposium, 2004..

[85] Koenen R., Overview of the MPEG-4 standard, ISO/IEC JTC1/SC29/WG11 N1730, Luglio Italy 1997

[86] Dranove D., Gandal N. , Surviving a standards war: lessons learned from the life and death of DIVX, Discussion Paper No. 3935, Centre for Economic Policy Research. London, 2003.

[87] Xvid features, at Xvid developers web site. Available at: http://www.xvid.org/Xvid-Codec.2.0.html (last accessed on 14/5/2018)

[88] Richardson I.E., "H. 264 and MPEG-4 video compression: video coding for next generation multimedia". Wiley Ed. 2004.

[89] Lloret J., Mauri P. V., Jimenez J. M., Diaz J. R., 802.11 g WLANs design for rural environments video-surveillance, In proceedings of the ICDT'06 International Conference on Digital Telecommunications, 2006. Cap Esterel (France). August 29-31, 2006, (pp. 23-23). DOI: 10.1109/ICDT.2006.1.

[90] Jiménez J. M., Díaz J. R., Sendra S., Lloret J., Choosing the Best Video Compression Codec Depending on the Recorded Environment. In proceedings of the 2014 IEEE Global Communications Conference (GLOBECOM 2014), Austin, TX (USA), December 8-12, 2014. (pp. 1347-1352). DOI: 10.1109/GLOCOM.2014.7036995.

[91] Ohwovoriole E., Andreopoulos Y., Comparison of Google/WebM VP8, AVC/H. 264 and TMuC HEVC, C TMuC," Proc. London Communications Symposium (LCS), Sep 2010, pp. 1-4

[92] Lambert P., De Neve W., De Neve P., Moerman I., Demeester P., Van de Walle R., Rae-distorstion performance of H. 264/AVC compared to

state-of-the-art video codecs,IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, Issue 1, Jan 2006, pp. 134,140

[93] Peter Deutsch L., Ogg DEFLATE Compressed Data Format Specification version 1.3. RFC 1951. May 1996.

[94] Daoust F., Hoschka P., Patrikakis C. Z., Cruz R. S., Nunes M. S., Salama Osborne D., Toward video on the web with HTML5, W3C/ERCIM, Sophia-Antipolis , France. October 2010

[95] Larbier. P., Using 10-bit AVC/H.264 encoding with 4:2:2 for Broadcast Contribution, ATEME, Bievres, France Mar.2003, Doc. JVT-G050

[96] Álvarez A., Pozueco L., Cabrero S., García Pañeda X., García R., Melendi D., Díaz Orueta G., Adaptive Streaming: A subjective catalog to assess the performance of objective QoE metrics, Network Protocols and Algorithms, Vol 6, No 2 (2014). Pp. 123-136

[97] Theora video compression. Available at: https://theora.org/ (last accessed on 14/5/2018).

[98] The Ogg container format. Available at: https://www.xiph.org/ogg/ (last accessed on 14/5/2018).

[99] ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services, November 2007

[100] WebM: an open web media project. Available at: https://www.webmproject.org/ (last accessed on 14/5/2018)

[101] Urry J. Climate change and society, Why the social sciences matter. pp. 45-59. 2015. Palgrave Macmillan, London. DOI: https://doi.org/10.1057/9781137269928_4

[102] Garcia M., Lloret J., Edo M., Lacuesta R., IPTV distribution network access system using WiMAX and WLAN technologies, In Proceedings of the 4th edition of the UPGRADE-CN workshop on Use of P2P, GRID and agents for the development of content networks, Munich (Germany), June 11 - 13, 2009. (pp. 35-44). Doi:10.1145/1552486.1552513

[103] Ferman A.M., Tekalp A.M., Mehrotra R. “Robust color histogram descriptors for video segment retrieval and identification”. IEEE Transactions on Image Processing, Vol.11, No.5, pp.497-508. May 2002. DOI: 10.1109/TIP.2002.1006397

[104] Lee J., Automatic prefilter control by video encoder statistics, Electronic Letters, Vol.38, No.11, pp.503-505. 2002. DOI: 10.1049/el:20020361

[105] Bahri N., Grandpierre T., Ayed M. A. B., Masmoudi N., Akil M., “Embedded real-time H264/AVC high definition video encoder on TI’s KeyStone multicore DSP”, Journal of Signal Processing Systems, vol. 86, Iss. 1, pp. 67-84, 2017. https://doi.org/10.1007/s11265-015-1098-x

[106] Zinner T., Hohlfeld O., Abboud O., Hossfeld T. Impact of frame rate and resolution on objective QoE metrics, In proceedings of the Second International Workshop on Quality of Multimedia Experience (QoMEX 2010), Trondheim (Norway) June 21-23, 2010. DOI: 10.1109/QOMEX.2010.5518277

[107] Neogi A. and Chiueh T., Compression techniques for active video content, In Proceedings of the Data Compression Conference (DCC 2002), April 2-4, 2002. Snowbird, UT, (USA). (p. 466-472) DOI: 10.1109/DCC.2002.1000009

[108] Ohm J-R., Advances in scalable video coding, Proceedings of the IEEE 93.1 (2005): 42-56.

[109] Vidal L. G., Procesamiento de video utilizando la plataforma DAVINCI, PhD Thesis. Instituto Politécnico Nacional- Centro de Investigación y Desarrollo de Tecnología Digital. June 2011.Tijuana, B.C.(Mexico)

[110] Alvarez-Flores E. P., Ramos-Munoz J. J., Navarro-Ortiz J., Ameigeiras P., Lopez-Soler J. M., User-Level Quality Assessment of a Delay-Aware Packet Dropping Scheme for VoIP, Network Protocols and Algorithms (2011). Vol.3, Issue 3, pp.38-66.

[111] Lloret J., Garcia M., Ferrer A. J., Morant M., Editing and Compressing HDTV videos for storage and transmission over Internet, WSEAS Transactions on Signal Processing. Vol. 2, Issue 10, Pp. 1379-1386, October 2006

[112] De Simone F., Ticca D., Dufaux F., Ansorge M., Ebrahimi T., A comparative study of color image compression standards using perceptually driven quality metrics, In Optical Engineering+ Applications . International Society for Optics and Photonics. August, 2008, pp. 70730Z-70730Z

[113] Dumic E., Grgic S., Grgic M., Comparison of HDTV formats using objective video quality measures. Multimedia tools and applications, (2010), Vol. 49, Issue 3, pp.409-424

[114] Doñate R., Lloret J., Jiménez J. M., Estudio de Códecs de Compresión MPEG-4 para su Aplicación a la Videovigilancia. XX

Simposium Nacional de la Unión Cientifica Internacional de Radio (URSI2005), Gandia (Valencia). 2005

[115] Wang Z., Lu L., Bovik A. C., Video Quality Assessment Based on Structural Distortion Measurement, Signal processing image communication", January 2004, Vol.19, Iss. 1, pp. 121-132

[116] Schwarz H., Marpe D., Wiegand T., Overview of the Scalable Video Coding Extension of the H.264/AVC Standard, IEEE Transactions on circuits and systems for video technology,September 2007, Vol.. 17, Iss. 9, pp.1103-1120

[117] Hu, F., Hao, Q., Bao, K., A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation, Communications Surveys & Tutorials, IEEE (Volume: 16, Issue: 4), 22 May 2014, pp. 2181 – 2206. http://dx.doi.org/10.1109/COMST.2014.2326417

[118] Scott-Hayward, S., O'Callaghan, G., Sezer, S., SDN Security: A Survey, IEEE SDN for Future Networks and Services (SDN4FNS 2013), 11-13 Nov. 2013, Trento, Italy, pp. 1-7. http://dx.doi.org/10.1109/SDN4FNS.2013.6702553

[119] Kreutz, D., Ramos, F.M.V., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., Uhlig, S., Software-Defined Networking: A Comprehensive Survey, Proceedings of the IEEE, Volume 103, Issue 1, Jan. 2015, pp. 14-76. http://dx.doi.org/10.1109/JPROC.2014.2371999

[120] Kaur, K., Singh, J., Ghumman, N. S., Mininet as Software Defined Networking Testing Platform, International Conference on Communication, Computing & Systems (ICCCS. 2014), 20 Feb - 21 Feb 2014, Chennai, India

[121] Paasch, C., Ferlin, S., Alay, O., Bonaventure, O., Experimental Evaluation of Multipath TCP Schedulers, ACM SIGCOMM Capacity Sharing Workshop (CSWS), 2014. ACM, August 18, 2014, Chicago, IL, USA. http://dx.doi.org/10.1145/2630088.2631977

[122] de Oliveira, R.L.S., Shinoda, A.A., Schweitzer, C.M., Rodrigues, P. L., Using Mininet for Emulation and Prototyping Software-Defined Networks, IEEE Communications and Computing (COLCOM 2014), 4-6 June 2014, Bogota, Colombia, pp. 1-6. http://dx.doi.org/10.1109/ColComCon.2014.6860404

[123] Keti, F., Askar, S., Emulation of Software Defined Networks Using Mininet in Different Simulation Environments, 6th International

Conference on. IEEE Intelligent Systems, Modelling and Simulation (ISMS 2015), Kuala Lumpur, Malaysia, 2015. p. 205-210. http://dx.doi.org/10.1109/ISMS.2015.46

[124] Azizi, M., Benaini, R., Ben Mamoun M., Delay Measurement in Openflow-Enabled MPLS-TP Network, Modern Applied Science, Vol 9, No 3, 2015, Canadian Center of Science and Education, January 10, 2015. http://dx.doi.org/10.5539/mas.v9n3p90

[125] Gupta, M., Sommers, J., Barford, P. Fast, accurate simulation for SDN prototyping, 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 12-15 August, 2013, Hong Kong. http://dx.doi.org/10.1145/2491185.2491202

[126] Panwaree, P., Kim, J., Aswakul, C., Packet Delay and Loss Performance of Streaming Video over Emulated and Real OpenFlow Networks, The 29th International Technical Conference on Circuit/Systems Computers and Communications (ITC-CSCC), Phuket, Thailand, July 1-4, 2014

[127] Megyesi, P., Botta, A., Aceto, G., Pescapè, A., Molnár, S., Available Bandwidth Measurement in Software Defined Networks, SAC 2016,April 04 - 08, 2016, Pisa, Italy. http://dx.doi.org/10.1145/2851613.2851727

[128] Noghani, K. A., M. Oguz Sunay, M., Streaming Multicast Video over Software-Defined Networks, IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2014), 28-30 Oct. 2014, Philadelphia, PA, USA, pp. 551 – 556. http://dx.doi.org/10.1109/MASS.2014.125

[129] Jarschel, M., Wamser, F., Hohn, T., Zinner, T., Tran-Gia, P., SDN-based Application-Aware Networking on the Example of YouTube Video Streaming, 2013 Second European Workshop on Software Defined Networks (EWSDN 2013), Oct. 10 - 11, 2013, Berlin, Germany, pp: 87-92. http://doi.ieeecomputersociety.org/10.1109/EWSDN.2013.21

[130] Salsano, S., Ventre, P. L., Prete, L., Siracusano, G., Gerola, M., Salvadori, E., OSHI-Open Source Hybrid IP/SDN networking (and its emulation on Mininet and on distributed SDN testbeds)., 2014 Third European Workshop on Software Defined Networks (EWSDN), IEEE, 1-3 September 2014, Budapest, Hungary, p. 13-18. http://dx.doi.org/10.1109/EWSDN.2014.38

[131] Lantz, B., Heller, B., McKeown, N. A network in a laptop: rapid prototyping for software-defined networks. 9th ACM SIGCOMM Workshop on Hot Topics in Networks, ACM, 20 – 21 October, 2010, Monterey, CA, USA, Article No. 19. http://dx.doi.org/10.1145/1868447.1868466

[132] IEEE Standards Association. Available at: http://standards.ieee.org/ (last accessed on 14/5/2018)

[133] Fàbrega L., Jové T., Network schemes for TCP elastic traffic in the Internet, Network Protocols and Algorithms, Vol 5, No 3 (2013), pp. 33-67

[134] Screen Digest, Study on the Socio Economic Dimension of the Unauthorized Use of Signals: Part I: Current Market and Technology Trends in the Broadcasting Sector, SCCR/19/12, OMPI, Geneva, December 2009.

[135] M. Atenas, S. Sendra, M. Garcia, J. Lloret, IPTV Performance in IEEE 802.11 n WLANs, IEEE Global Communications Conference (IEEE Globecom 2010), Miami (USA), December 6 – 10 , 2010. pp. 929-933

[136] Hossfeld, T.,Leibnitz, K., A qualitative measurement survey of popular Internet-based IPTV systems, Second International Conference on Communications and Electronics, (ICCE 2008), Hoi An (Vietnam), 4-6 June 2008, pp. 156-161

[137] Cisco VNI website. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/vni-service-adoption-forecast/Cisco_VNI_SA_Forecast_WP.html (last accessed on 14/5/2018)

[138] Hernández L. ,Baladrón C. , Aguiar J. M. , Calavia L. , Carro B. , Sánchez-Esguevillas A. , Pérez F. , Fernández Á. , Lloret J. . Artificial Neural Network for Short-Term Load Forecasting in Distribution Systems, Energies, vol. 7, no. 3, pp. 1576-1598, 2014. 10.3390/en7031576.

[139] Garcia M., Sendra S., Turro C., Lloret J., User's Macro and Micro-mobility Study using WLANs in a University Campus, International Journal On Advances in Internet Technology, Vol. 4, Issue 1&2, Pp. 37-46 July 2011.

[140] Edo M., Garcia M., Turro C.,Lloret J., IP Telephony development and performance over IEEE 802.11g WLAN, The Fifth International Conference on Networking and Services (ICNS 2009), Valencia (España), April 20-25, 2009.

[141] Canovas A., Boronat F., Turro C., Lloret J., Multicast TV over WLAN in a University Campus Network, 5th Int. Conf. on Networking and Services, Valencia, April 20-25, 2009.

[142] Granell E., Bri D., Tomas J., J. Lloret, Smart Devices Fingerprint Detection, IEEE Global Communications Conference 2012, Anaheim, California (USA), December 3 – 7, 2012.

[143] Canovas A., Bri D., Sendra S., Lloret J., Vertical WLAN Handover Algorithm and Protocol to Improve the IPTV QoS of the End User, IEEE International Conference on Communications 2012, Ottawa (Canada), June 10–15, 2012.

[144] Sadiq A. S., Bakar K. A., Ghafoor K. Z., Lloret J., Mirjalili S. A., A Smart Handover Prediction System based on Curve Fitting Model for Fast Mobile IPV6 In Wireless Networks, International Journal of Communication Systems.

[145] Sendra S., Fernández P.A., Quilez M.A., Lloret J., Study and Performance of Interior Gateway IP Routing Protocols, Net. Prot. and Alg., Vol. 2, Is. 4, Pp. 88-117. Dec. 2010.

[146] Silverston T., Fourmaux O., Measuring P2P IPTV Systems, Proc. of ACM NOSSDAV, Urbana-Champaign, Illinois, USA, June 4-5, 2007.

[147] Hjelmvik E., John W., Statistical protocol identification with spid: Preliminary results, Swedish National Computer Networking Workshop 09, Uppsala, Sweden, May 4-5, 2009

[148] Lloret J., Garcia M., Bri D., Diaz J. R., Study and performance of a group-based content delivery network, Journal of Network and Computer Applications. Vol. 32, Issue 5, September 2009, pp. 991-999.

[149] Lloret J., Garcia M., M. Atenas, Canovas A., A Stereoscopic Video Transmission Algorithm for an IPTV Network Based on Empirical Data, International Journal of Communication Systems, Vol. 24, Issue 10, October 2011, pp. 1298–1329.

[150] Vapnik V.N., An overview of statistical learning theory, IEEE Transactions on Neural Networks, vol.10, no.5, Sep 1999, pp. 988-999

[151] Thomas R. W., Friend D. H., Dasilva L. A., Mackenzie A. B., Cognitive networks: adaptation and learning to achieve end-to-end performance objectives, IEEE Communications Magazine, vol.44, no.12, Dec. 2006, pp.51-57.

[152] Gürer D. W., Khan I., Ogier R., Keffer R., An Artificial Intelligence Approach to Network Fault Management, SRI International, Menlo Park, California, USA

[153] Lloret J., Atenas M., Canovas A., Garcia M., A network management algorithm based on 3D coding techniques for stereoscopic IPTV delivery. IEEE Global Communications Conference 2011, Houston, Texas, USA, December 2011

[154] Lloret J., Garcia M., Tomas J., Sendra S., A Group-based Architecture for Grids, Telecommunication Systems, Vol. 46 Issue 2, February 2011, pp. 117-133, 2011

[155] Canovas A., Garcia M., Lloret J. , Atenas M., Rizo R., Improving IPTV QoE taking the suitable MPEG-2/MPEG-4 Quantizer based on Jitter, Delay and lost packets measurements, Int. Journal on Advances in telecommunications, Vol. 3, Issue 3 &4, December 2010, pp. 129 - 139.

[156] MATLAB. Math. Graphics. Programming. Available at: https://www.mathworks.com/products/matlab.html (last accessed on 14/5/2018)

[157] Garcia M., Canovas A., Edo M., Lloret J., A. QoE management system for ubiquitous IPTV devices, The Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM 2009. October 11-16, 2009, Sliema, Malta.

[158] Jimenez J. M., Romero O., Rego A., Dilendra A., Lloret J., Study of Multimedia Delivery over Software Defined Networks, Network Protocols and Algorithms, Vol 7, No 4 (2015), pp. 37-62

[159] Recomendation UIT/T G.1010: End-user multimedia QoS categories (2001). Available at: https://www.itu.int/rec/T-REC-G.1010-200111-I/en (last accessed on 14/5/2018)

[160] Recomendation UIT‑R M.1079-2: Performance and quality of service requirements for International Mobile Telecommunications-2000 (IMT-2000) access networks. Available at: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1079-2-200306-I!!PDF-E.pdf (last accessed on 14/5/2018)

[161] Cisco Systems Inc. Cisco's Visual Networking Index Forecast Projects Nearly Half the World's Population Will Be Connected to the Internet by 2017, Available at: http://newsroom.cisco.com/press-release-content?articleId=1197391 (last accessed on 14/5/2018)

[162] RTP: A Transport Protocol for Real-Time Applications. Available at: https://www.ietf.org/rfc/rfc1889.txt (last accessed on 14/5/2018)

[163] An Instant Virtual Network on your Laptop (or other PC). Available at: http://mininet.org/ (last accessed on 14/5/2018)

[164] CISCO. "Support – Cisco 2800 Series Integrated Services Routers (Cisco 2811 Integrated Services Router)". Available at: http://www.cisco.com/c/en/us/support/routers/2811-integrated -services-router-isr/model.html (last accessed on 14/5/2018)

[165] CISCO. "Support – Cisco Catalyst 3560 Series Switches (Cisco Catalyst 3560-24PS Switch)". Available at: http://www.cisco.com/c/en/us/support/switches/catalyst-3560-24ps-switch/model. html (last accessed on 14/5/2018)

[166] Videolan software. Available at: http://www.videolan.org/vlc/ (last accessed on 14/5/2018)

[167] Wireshark software. Available at : https://www.wireshark.org/ (last accessed on 14/5/2018)

[168] NS-3 (Online article). Available at: https://www.nsnam.org/overview/key-technologies/ (last accessed on 14/5/2018)

[169] EstiNet Technologies Inc. Available at: http://www.estinet.com/index.php (last accessed on 14/5/2018)

[170] The OpenDaylight Platform. (Online article), Available at: https://www.opendaylight.org/ (last accessed on 14/5/2018)

[171] Component-based software defined networking framework. (Online article) Available at: http://osrg.github.io/ryu/ (last accessed on 14/5/2018)

[172] Project Floodlight. (Online article) Available at: http://www.projectfloodlight.org/floodlight/ (last accessed on 14/5/2018)

[173] VLC media player Website. Available at: http://www.videolan.org (last accessed on 14/5/2018)

[174] Wireshark website. Available at: https://www.wireshark.org/ (last accessed on 14/5/2018)

[175] Barambones, C., Pascual, D., Diaz, J. R., Lloret, J., Una Nueva Herramienta para Testear el Rendimiento de una Red IP, XI Jornadas de Ingeniería Telemática (JITEL 2013), Granada (Spain), October 28-30, 2013.

[176] Big Buck Bunny website. Available at: http://www.bigbuckbunny.org (last accessed on 14/5/2018)

[177] Downline Guide, Big Buck Bunny, Sunflower version. Available at: http://bbb3d.renderfarming.net/download.html (last accessed on 14/5/2018)

[178] FFmpeg. Available at: http://www.ffmpeg.org/about.html (last accessed on 14/5/2018)

[179] Sullivan G. J., Wiegand T., Video Compression - From Concepts to the H.264/AVC Standard. Proceedings of the IEEE, Vol. 93, No. 1, (2005)

[180] tPacketCapture. Available at: https://play.google.com/store/apps/details?id=jp.co.taosoftware.android.packetcapture&hl=es (last accessed on 14/5/2018)