POLITECNICO DI TORINO

Master Degree Course in Computer Enginering

Master Degree Thesis

Design And Development Of An Architecture For Real-time Enhancement Of TV Streams

Supervisor

prof. Paolo Prinetto

Candidate

Nicolò Maunero

December 2017

Acknowledgements

I would first like to thank my thesis advisor Prof. Paolo Prinetto and Dott. Giuseppe Airò Farulla that guided me through the whole process of this thesis providing great support and wise advice. I would also like to thank RAI - Centro Ricerche e Innovazione Tecnologica, Unità Tecnologie per Accessibilità Prodotto RAI e Tecnologie per la e-Inclusion team for allowing me to work with them in an always friendly and motivating environment. I must express my very profound gratitude to my parents and a special thanks to my brother Emanuele for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Contents

Chapter 1

General Introduction

This Thesis takes place in a bigger and wider project called Stretch TV, started by RAI – Centro Ricerche e Innovazione Tecnologica and by RAI – Responsabilità Sociale. Its main aim is to design and develop some of the functionalities required by the Stretch TV project such as the television streaming enhancement as well as a way to remotely control the platform.

Cinema and television have been experienced an extraordinary evolution both in terms of technical means and in their contents.
The need of keep up with a society with more and more tightened pace of life has changed deeply the way in which television programs have been structured and realized.
The way in which television programs are hosted has radically changed from the 50's till today, from a black and white television that transmits only in certain hours of the day, to a TV scheduling like the actual one with channels that transmits 24/7 and characterized by fast rhythm and short time.
Also the user has been changed, today the televiewer rapidly selects among a huge number of different channels with the remote controller in order to decide what to watch, and with the Smart TV connected to the net, it can be able to select within a vast selection of different contents in order to define its own and personal TV scheduling.
Elder and people affected by cognitive or sensorial disability, on the other hand, ask for a reduction in the complexity and technological barriers of the television medium.
The Stretch TV project, hence, aims to develop a technological solution to greatly improve the usability of the television contents among people with sensorial and/or cognitive disability as well as elders.

Considering the target people to which the Stretch TV project aims for, one of the key aspects to be taken into account is the speech speed. In fact, elders or in general people affected by hearing impairment, may found difficult and hard to understand television programs talks, especially in those programs characterized by an high speech rate. Since the birth of the television as a consumer goods for the industrialized world, it has been possible to notice how there was a progressive increase on the speech speed, mostly in programs like TV news.

Considering TV news, maybe just for the necessity to give to the viewers a constantly increasing quantity of information from around the world in an reduced amount of time, we can notice how there was an incredible increase in the speech speed of the program host or journalists.
Using as a reference a study case performed by " RAI - Centro Ricerche e Innovazione Tecnologica, Unità Tecnologie per Accessibilità Prodotto RAI e Tecnologie per la e-Inclusion " taking as a sample an excerpt of a TV news from around the late 60's, it can be noticed by a qualitative measuring that the speech speed is around 118 word per minute, while in a TV news today the program host or journalist speech speed can reach 180/200 even almost 210 word per minute.
This high speech speed, for a normal person may not be a problem, but for people with sensorial or cognitive disability and elders can be extremely hard to keep up with the talk, making frustrating and almost impossible for them to understand. Hence a way of reducing the speed of the television multimedia streaming can come in handy, making possible to these people to adjust television programs to a speed they are comfortable with and giving them the possibility to easily understand and watch those programs.

Another important problem to be taken into account while defining which are the key features of the Stretch TV project is the hearing loss.
This problem can afflict everybody, but is especially evident and mostly spread among elders (in this case is denoted as presbycusis [1]).
The hearing loss can be caused by genetic factor and/or damage of the auditory apparatus and it shows itself into a total or partial loss of the hearing aids.
Almost 30 - 35% of adults of age between 60 and 75 year are affected by this disorder, while it is estimated that 40 - 45% of adults of age greater than 75 years are affected by this disorder [2]. The presbycusis is the ensemble of auditory apparatus problems that are caused by the normal aging of the apparatus itself, it can be noticed as a progressive difficulty or inability to hear some of the frequencies of the auditory spectrum that can be more or less severe and it affects mostly higher frequencies with respect to low tones.

[1] Studio Di Otorinolaringoiatria Tanzariello, 2016. *Presbiacusia.* Available at: http://www.tanzariello.it/index.php/orecchio/patologia-orecchio/orecchio-medio/145-studio-prof-a-tanzariello/orecchio/patologia/orecchio-interno/ipoacusia-neurosensoriale-dell-adulto-di-tipo-progressivo/781-presbiacusia. Accessed 6 November 2017

[2] EAD. *Presbiacusia*

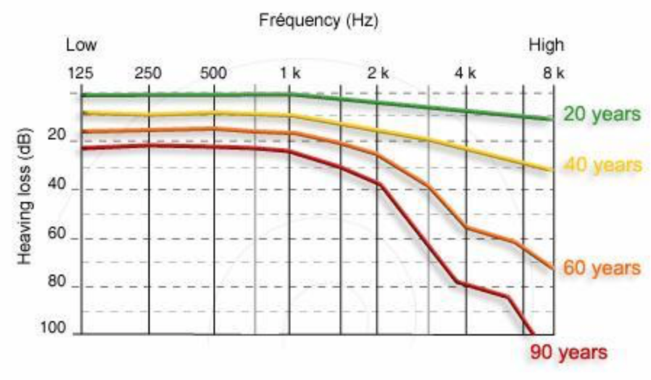


Figure 1.1. Audiogram Example of Presbycusis at Different Ages [3]

In Figure 1.1 is shown a typical audiogram of presbycusis at different ages. It shows how the hearing threshold, especially for high frequencies tones, may vary with advancing age, in particular as a person grown old the hearing threshold become worse. Hence a system able to equalize television streaming audio, going to emphasize those frequencies on which the hearing is more lacking, can greatly help elders, and all people that are affected by these disturbs, to easily listening to television programs.

The Stretch TV project aims to integrate both the capability of slowing down audio and video streaming as well as the audio equalization without affecting the intelligibility of the speech to help a wide variety of people to better understand and consume the television medium. This system can be used in a wide variety of applications ranging from comprehension of television programs and cognitive rehabilitation process to learning the Italian language as well as foreign ones.
The system developed for the Stretch TV project is composed of an hardware platform connected to the antenna signal and to a screen. This allows the acquisition of the television signal, the manipulation of it using a software component and the display of the modified streaming. Several actions can be performed, from the basic possibility to tune among different television channels to the possibility for the final user to change the speed of the reproduced video and choose among different audio equalization options, to watch the television show in the most suited manner, moreover the capability of a live replay is also made available. The objective of this Thesis is to develop an effective way of communication with the platform, in particular by enabling the possibility of controlling the system remotely using a smartphone or tablet. In addition, an architecture for performing system updates on the platform has been introduced, giving the possibility of updating

[3] Studio Di Otorinolaringoiatria Tanzariello, 2016. *Presbiacusia.* Available at: http://www.tanzariello.it/index.php/orecchio/patologia-orecchio/orecchio-medio/145-studio-prof-a-tanzariello/orecchio/patologia/orecchio-interno/ipoacusia-neurosensoriale-dell-adulto-di-tipo-progressivo/781-presbiacusia. Accessed 6 November 2017

the system without the needs of a central server managing this task. Finally some modification and adjustment to the platform are been made in order to correct some error and improve the management of multimedia processing processes to guarantee a more stable and usable environment.

This document is structured as follows: the first chapter gives a general introduction about the Stretch TV project, the reason behind its design and development and which are the main groups of people this project tries to address. The second chapter gives a brief overview over some of the commercial off-the-shelf available, that are similar or somewhat related to the project itself.
The third chapter gives a presentation of the video processing libraries available for developers and their weakness and strength. The fourth chapter gives a presentation of the architectures previously analyzed and researched to foster the development of the Stretch TV platform as well as some of the motivations about why they were considered in the first place and why they were abandoned after.
The fifth chapter gives a more in depth presentation about what are the constraints and requirements for the development of the Stretch TV platform and an identification of which are the possible hardware and software resources to be used.
The sixth chapter gives a brief presentation and overview about the platform used for the Stretch TV project, how it is composed and how it works. The seventh chapter gives a presentation of some steps of the design process of this Thesis work. The eighth chapter gives a presentation on how the implementation of the works of this Thesis has been carried out and some of the reason behind the decision made.
The ninth chapter gives a report of performance tests performed on the platform and an analysis of the results obtained. Finally, the tenth chapter gives a brief summary on the Stretch TV project, some considerations about it and which are the possible future developments and improvements of the project itself.

Chapter 2

State of The Art

Given the great growing of web based streaming services, such as Netflix, Amazon Prime Video and many others, and television content on-demand, the necessity and will of developing a platform capable of manipulate the live television streaming, receiving it directly from the antenna signal, are become less and less important and requested by the consumers.
There exist different hardware solution that allows to transform an old television into a smart TV (such as the great quantity and diversity of android TV boxes), but what those kind of devices do is to simply give the possibility to access the more common web streaming platforms. Instead, solution similar to a TV box but that allows the capture and manipulation of the signal received directly from the antenna, are poorly present on the market.
Some of the principal solutions available are reported below.

2.1 TVCenter

Figure 2.1. TvCeter

TVCenter [1] is a software application produced and developed by Pinnacle, it requires the uses of PCTV USB stick[2], also produced by Pinnacle, in order to access television content.
The PCTV USB stick is DVB-T tuner, it enables the capability of tuning to the desired television channel; it has to be connected to the antenna signal on one side and on the other side to the personal computer, and, in conjunction with the TVCenter software, gives the possibility to reproduce on the computer the television contents.
This platform support a wide variety of functionalities, from the possibility of watching live TV contents to the possibility of recording them and watching later. It supports also, only for recorded contents, the selection of different audio tracks and subtitles.
It does not support video content reproduction in slow motion.

2.2 EyeTV

Figure 2.2. EyeTV

EyeTV is a DVB-T tuner with a dedicated software application, initially developed and produced by Elgato[3], has now been moved to Geniatech[4].

[1] http://www.pctvsystems.com/TVCenter6/tabid/207/language/it-IT/Default.aspx

[2] http://www.pctvsystems.com/Products/ProductsEuropeAsia/DVBTT2products/tabid/141/language/it-IT/Default.aspx

[3] https://www.elgato.com/en/eyetv

[4] https://www.geniatech.eu/eyetv

The EyeTV system is composed of an DVB-T USB stick tuner that can be connected to the antenna and is used to perform the tuning among all the available television channel. On the other side, the system comprehends also a software application from which the user can access the television content and control the USB tuner.
With the EyeTV software it is possible to watch live TV on the computer as well as record television programs. It provides a wide variety of features like pause and rewind live TV programs, setup a schedule for the recording and many others. One interesting feature is the possibility, only for recorded content, to perform the video reproduction in slow motion, the drawback is that no audio equalization is performed within this feature and so the speech results distorted.

Summarizing, considering the generic topic of technologies developed for elders and people affected by cognitive and/or sensorial disability, a wide variety of product and studies can be found in medicine and in scientific research. Those type of products can ranging from cochlear implants for helping hearing impaired people, to assistive technologies for rehabilitation and ad-hoc audiovisual contents for people affected by cognitive and/or sensorial disability. Considering the specific case of the Stretch TV project instead, we can not directly rely on of-the-shelf custom solutions, all the available platform do not perform any adjustment or include any features addressing elders or people with sensorial and/or cognitive disability, hence the development of a platform with this capabilities can surely be helpful and appreciated.

Chapter 3

Audio And Video Processing Libraries

3.1 FFMPEG

Figure 3.1. FFmpeg Logo

FFmpeg[1] is the leading multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter and play pretty much any audiovisive format. It support a wide variety of formats up to the cutting edge, no matter if they were designed by some standard committee, the community or a corporation. It is also highly portable: FFmpeg compiles and runs across Linux, Mac OS X, Microsoft Windows, the BSDs, Solaris, etc. under a wide variety of build environments, machine architectures, and configurations.
The FFmpeg project tries to provide the best technically possible solution for developers of applications and end users alike. In order to achieve this its combine the best free software option available. The FFmpeg code is slightly favored to keep the dependancies on other libraries low and to maximize code sharing among different parts of FFmpeg.
FFmpeg is composed of different tools:

- **ffmpeg**: a command line tool convert multimedia files between formats, also capable of performing complex operation and manipulation.

[1] https://www.ffmpeg.org

- **ffplay**: a simple media player based on SDL and the FFmpeg libraries.
- **ffserver**: a multimedia streaming server for live broadcast.
- **ffprobe**: a simple multimedia stream analyzer.

Also different libraries are made available for developers:

- **libavutil**: is a library containing functions for simplifying programming, including random number generators, data structures, mathematics routines, core multimedia utilities, and much more.
- **libavcodec**: is a library containing decoders and encoders for audio/video codecs. FFmpeg support and utilize a wide variety of codecs, from mpeg (standard definition) and H264 (high definition) for viedo formats, to AAC3 and others for the audio.
- **libavformat**: is a library containing demuxers and muxers for multimedia container formats.
- **libavdevice**: is a library containing input and output devices for grabbing from and rendering to many common multimedia input/output software frameworks.
- **libavfilter**: is a library containing media filters. Filters are used to perform manipulation operation of the multimedia streaming. Several action can be performed from dividing audio and video traks to performing more complex operation like reducing the content playback speed to combining together different aduio/video traks and subtitles.
- **libswscale**: is a library performing highly optimized image scaling and color space/pixel format conversion operations.
- **libswresample**: is a library performing highly optimized audio resampling, rematrixing and sample format conversion operations.

Moreover security is a high priority for FFmpeg developers and code review is always done with security in mind, providing quick update when new security issue are found.
Overall FFmpeg is a complex framework able to perform many and different manipulation operation of multimedia streams, it is widely adopted and used in video manipulation tasks and modern media player softwares.

3.2 GStreamer

Figure 3.2. GStreamer Logo

GStreamer[2] is an open source multimedia framework. It works on all major operating system such as Linux. Android, Windows, Mac OS X, iOS, as well as most BSDs, commercial Unixes, Solaris and Symbian. It has been ported to a wide range of operating system, processors and compilers.
It runs on all major hardware architectures including x86, ARM, MIPS, SPARC and PowerPC, on 32-bit as well as 64-bit and little endian or big endian.
GStreamer can bridge to other multimedia frameworks in order to reuse existing components, such as codecs, and use platform input and output mechanism.

- **Linux/Unix**: OpenMAX-IL (via gst-omx)
- **Windows**: DrectShow
- **Mac OS X**: QuickTime

It is constructed around a good, stable and compact (less than 500KB) core library containing and making available to developers many features and tools for a quick and easy development in a wide variety of contexts and applications.
A graph-based structures allows for arbitrary and easy pipeline construction, moreover allows for a more trivial and transparent multi-threaded pipeline construction, it is based on GLib 2.0 object model[3] for object-oriented design and inheritance, data passing is made extremely lightweight in order to allows very high performance and low latencies. Internal clocking system ensures global inter-stream synchronization (audio and video synchronization).
The GStreamer multimedia framework structure is based around the concept of plugins. Plugins are modules that can be seamlessly incorporated into GStreamer providing all kinds of elements and media type such as audio and video codecs, various container formats, web-based streaming capabilities and many other functionalities for audio and video manipulation. Furthermore, these plugins are dynamically loaded and resource friendly, they are developed following the strict requirement of not to waste memory; another interesting feature of plugins is the possibility to visualize, by dumping them to a .dot file

[2] https://gstreamer.freedesktop.org

[3] http://developer.gnome.org/gobject/stable/

and creating a PNG image from that, the pipelines developers have created for a better visualization of the overall development process.

Summarizing, GStreamer is a complex multimedia framework, capable of managing many audio and video formats and codecs, manipulating and doing different kind of operation on a wide variety of multimedia elements; its intelligent plugins architecture allows for a great modularity and versatility and can be used in a great number of different applications and architectures. It provides also a command-line tool for quick prototyping and testing as well as a large integration of its own API with various programming languages.

3.3 Sonic

Sonic[4] is a simple algorithm for speeding up or slowing down speech. The Sonic library is a very simple ANSI C library that is designed to be easily integrated into streaming voice application. The primary motivation behind Sonic is to enable the blind and visually impaired to improve their productivity with open source speech engines, like espeak, but this library can also be used for helping hearing impaired people. For example, Sonic can improve the experience of listening to an audio book on an Android phone.

A native Java port of this library is also available in order to increase the possibility if integration in many software.

Sonic also integrate a pitch change algorithm in order to not obtain distorted audio when speeding op or slowing down.

This solution only performs speech rate manipulation and an integration with a video streaming component is not present.

[4] https://github.com/waywardgeek/sonic

Chapter 4

Designed Architectures

The main goal of the Stretch TV project is to provide a platform to be integrated with a normal television and that is capable of provides all the required functionalities. The primary task to be achieved is to have a platform that combines both hardware and software for receiving the television signal directly form the antenna, manipulate this multimedia streaming following the requests and preferences of the user and offering the manipulated audiovisual content to the user to guarantee the best possible experience in the fruition of the television entertainment.
Different solution have been analyzed assessing pros and cons with respect to what are the necessities of the Stretch TV project.

4.1 USB Armory

USB Armory is an open source hardware design from Inverse Path[1] and it implement a flash drive sized computer (very compact form factor 65 x 19 x 6 mm).
It features an NXP i.MX53 ARM® Cortex™-A8 800Mhz as well as 512MB DDR3 RAM. Its hardware is also supported by standard software environments and requires very little customization efforts to be installed and used. In facts provides an excellent native support for all the major Linux distributions such as Debian, Ubuntu and Arch Linux as well as Android. Since the USB Armory does not have any video output connection capability, the platform has been designed taking advantage of a decoder, in particular the decoder receives the signal from the antenna, the USB Armory that is connected to the decoder, detect whether a multimedia streaming is being captured and provides to manipulate it according to the required modification to be performed, then the modified streaming of the USB Armory is written back to the decoder that send it back to the television for display, all this operation has to be done in real time, reading the input video streaming while is written by the decoder and writing back the modified one.
The main benefit of this approach is in the small form factor of the USB Armory computer

[1] https://inversepath.com/usbarmory.html

resulting in a very space-saving solution. The major problem of this platform, on the other hand, is the poor performance of the USB Armory computer, in fact it is produced for security applications and it not provide any GPU or hardware acceleration capabilities and it makes impossible to manipulate the multimedia streaming in real-time with enough speed for guarantee a seamless video output.

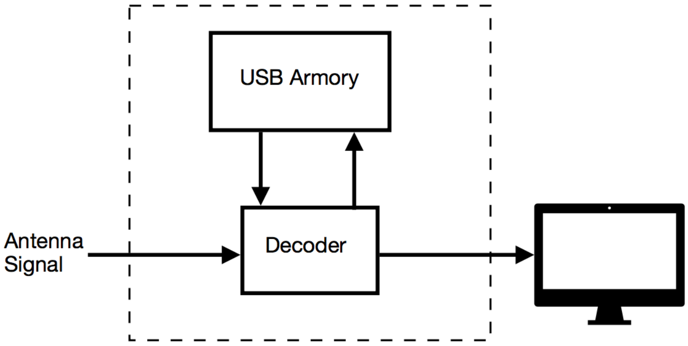


Figure 4.1. Platform Schematic involving use of USB Armory

Figure 4.2. Image of the USB Armory flash drive

4.2 Raspberry Pi

Raspberry Pi[2] is a well know, and widely adopted in many different applications, single board computer.
It is produced by the Raspberry Pi Foundation and comes in various products differentiated by their hardware specifications.
For the specific case the Raspberry Pi 3 has been selected, it features

- 1.2 GHz 64-bit quad-core ARM Cortex-A53
- 1 GB of RAM (shared with GPU)

[2] https://www.raspberrypi.org

- Broadcom VideoCore IV working at 400 MHz

Moreover this single board computer provides also a wide variety of port interfaces such as ethernet, various USB ports, HDMI as well as GPIOs and many others. Since the Raspberry Pi has some video output capability, it feature an HDMI output port, with respect to the USB Armory platform the architecture has been changed. The television live stream is provided to the Raspberry Pi board through a DVB-T tuner connected via USB that takes the signal directly from the antenna. The single board computer is in charge of manipulating the multimedia stream and send it to the television. This solution provides a very compact and low cost platform, but on the other hand, also in this case, even with the computing capabilities of the GPU core, it is not possible to produce on the output a seamless and acceptable video streaming.

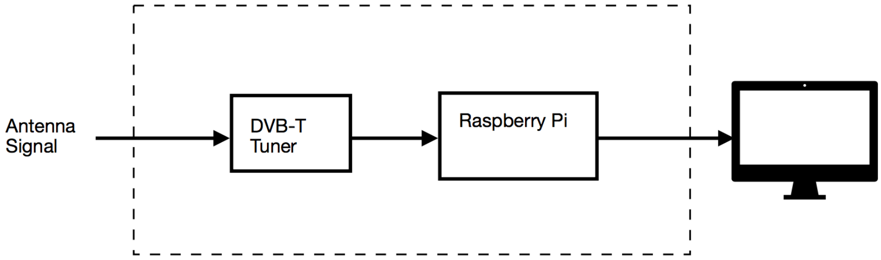


Figure 4.3. Platform Schematic involving use of Raspberry Pi

Figure 4.4. Image of the Raspberry Pi board

4.3 Smart TV

The third option taken into account for the Stretch TV project platform implementation flows around the use of a Smart TV.

Different kind of approach to the project needs were considered leading to diverse possible solutions. The first idea considered was to integrate the Stretch TV project as a features of the interactive Rai on-demand platforms; the television contents, thus, that are available

are transformed through a chain of digital post-processing in new contents slowed down and with characteristics of high comprehensibility.
Another possibility was to integrate the slowing down and audio equalization capabilities of the Stretch TV project as an integrated features of the Smart TV. Hence the multimedia content will be recorded by the television itself and proposed in a slowed down and equalized form to the final user.
The main positive aspect of using a Smart TV for the Stretch TV platform development is that through this services, especially the one involving the use of the Rai on-demand web streaming service, it is possible to provide those kind of multimedia content to a large amount of people, everyone that owns a Smart TV without the needs of any additional hardware components, and not only elders or people affected by disability. The principal drawback of this approach was the necessity to buy a Smart TV, for people that are not owners of one of it, that comes at an high cost and has to be taken into account that Smart TV ownership nowadays is not widely spread among population especially considering elders, that are one of the main targets of this project.
Surely this solution in future, when Smart TV ownership will be widely spread among population, shall be better considered.

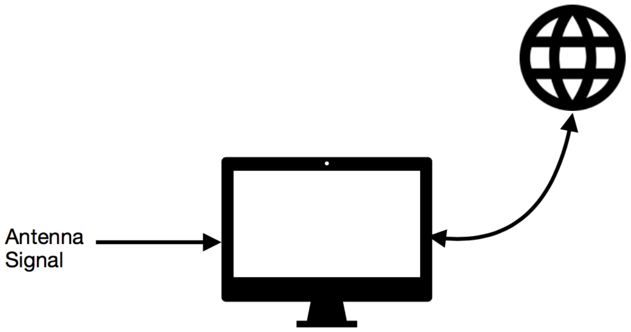


Figure 4.5. Platform Schematic involving the use of Smart Tv

Chapter 5

Hardware and Software Technologies Identification

Users and stakeholders require a system ready to be integrated with a normal television, able to receive the television streaming signal directly from the antenna, and through a specific software component, capable of manipulate that streaming in real time, in particular we are interested in the possibility of slowing down the multimedia streaming and apply the proper audio equalization techniques. Moreover some other features have to be added in order to make the use of the Stretch TV platform as similar as the normal use of a television, such as the possibility to tune in different channel and the possibility of an instant live replay of the multimedia content in order to make the overall use experience as appreciated as possible. Furthermore the overall platform has to be designed keeping in mind that this is a product primarily meant for elders or people affected by cognitive and/or sensorial disability so each component, especially the ones used for controlling the platform need to be simple and easy to use.

To make the interaction with the platform accessible to people, easy and straightforward, avoiding the needs of manually typing the various commands and operations to be performed, a user interface for controlling the system become essential and has to be designed. As a first step in this direction a simple web page was developed, it contains a list of the features proposed by the Stretch TV platform, such as the possibility to tune into few different channels, the selection of the reproduction speed as well as different audio equalizations and the possibility of an instant replay of the television streaming. The objectives of this Thesis, hence, was, on one hand, to design and develop a custom mobile application that introduced an architecture for remotely control the platform and that substituted the web page in order to make a step in the direction of simplifying the usability of the platform, that requires little effort and knowledge in it use and configuration and in order to mimic the possible use of a normal remote controller and television. In the application the functionalities proposed were expanded, adding more channels to tune in and more selections for the instant replay, providing to the user a simple and clear interface through which interact with the system. Design and development of an architecture for future updates of the platform software has been carried out and has been integrated as a built-in functionality of the mobile application avoiding the needs of a central server. On the other improve the overall usability of the Stretch TV platform, by correcting errors and by better managing multimedia processing processes, providing, hence, a more stable system.

A first step of the design process was to identify which are both hardware and software resources needed to fulfill the Stretch TV project platform requirement. Considering the hardware part first, what was necessary initially was a component able to capture the television streaming signal directly from the antenna and able to tune it giving the possibility of choosing within different channel to be displayed, for this specific task the PCTV nanoStick[1] has been selected; it is a small USB DVB-T tuner capable of receiving the digital terrestrial television signal. The second hardware component needed was a PC powerful enough to been able to manipulate the multimedia streaming in real time and producing as the output for the final user a seamless streaming content without loosing in quality or intelligibility.
For the streaming manipulation process, the FFmpeg multimedia framework has been selected, in fact it has been proven to be able to largely fulfill all the audio and video manipulation requirements of the Stretch TV project, providing the proper features and tools and many more.
For what concerns in the specific the work of this Thesis, hence, the remote control of the Stretch TV platform, and more in general the communication and connection within the remote controller (in this case a smartphone or tablet) and the system, the wireless communication that uses the Wi-Fi technology over the local network has been identified in order to been able to interconnect the remote controller with the platform and send the appropriate control signal to the platform in order to select the preferred operation to be done.

[1] http://www.pctvsystems.com/Products/ProductsEuropeAsia/DVBTT2products/PCTVnanoStick/tabid/167/language/it-IT/Default.aspx

Chapter 6

The Platform

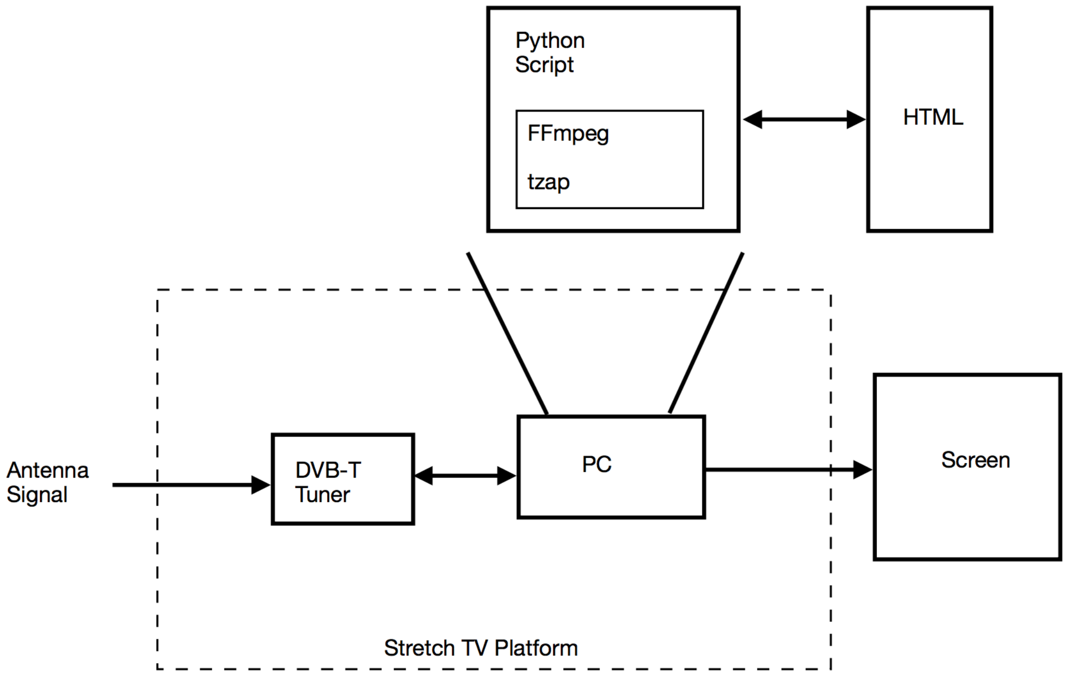


Figure 6.1. Stretch TV Platform Schematic

Figure 6.1 shows a schematic of the Stretch TV platform, as it was at the beginning of the work of this Thesis. Taking into consideration all the architectures previously analyzed and studied this one, in it form and basic concepts, does not move away too much. The idea and basic concepts behind, in fact, remain unchanged, while the principal components, in particular the computer that provides computations capacity for executing the proper manipulation operations on the multimedia streaming, have been changed to a more powerful components in order to been able to provide the proper computation capabilities and producing on the output a seamless video streaming without loosing in quality.

DVB-T tuner: this component is a USB stick capable of capturing the television streaming and providing it to the computer. DVB-T stands for Digital Video Broadcasting - Terrestrial and it is the European-based consortium standard for the broadcast transmission of digital terrestrial television and this system transmits compressed digital audio, digital video and other data in an MPEG transport stream.
Hence, the DVB-T USB stick, on one side is connected directly to the antenna cable receiving the television signal, on the other side it is connected to the computer providing to it the multimedia content. Moreover on the computer another tool concerning the DVB-T tuner is used. It is the LinuxTV dvb-apps package [1], it contains some application and a set of utilities, in the form of command line tools, that provides control capabilities over the DVB-T tuner, in fact through these utilities and API it is possible to scan for all the channel received from the antenna signal and tune among all this different channels.

Computer: the computer in question is a Mac Mini, this component is powerful enough to been able to perform all the multimedia streaming modifications and manipulations in real time. The computer features a dual core intel i5 CPU clocked at 2.6 GHZ and 8 GB of DDR3 RAM at 1600 MHz. The whole software part of the Stretch TV platform runs on a Linux Virtual Machine with Ubuntu 16.4 installed. This mainly due to lacks of drivers support for the DVB-T tuner from Apple and scarce support for all the FFmpeg tools and features for the Mac Os operative system.

Python Script: this is used to perform all the commands and issue the instructions required for FFmpeg and for controlling the USB DVB-T tuner. Thanks to FFmpeg is possible to manipulate both audio and video to obtain the desired result, while for controlling the USB DVB-T tuner tzap is used, it is one of the utilities contained in the LinuxTV dvb-apps package, and it allows to tune into the different television channels and record the multimedia streaming on the computer. The script contains the list of all the commands, functions and operations to be performed in order to effectively manipulate the multimedia streaming, such as the opening and closing of streaming channels, the management of audio and video flows, that are represented on the system by proper files, and all the other minor operations that are required for a correct and functional operation of the system.

HTML Page: a simple HTML page containing a list of some of the functionalities provided by the platform. It is a simple web page, accessible through the browser, that defines a preliminary user interface for controlling the system, but it is not meant to be used by customers. It provides to developers a way for selecting which are the operation to be performed strongly simplifying the interaction with the platform, avoiding the needs

[1] https://www.linuxtv.org

of manually typing the various commands. Defining, together with the python script mentioned before, the basis for a future remote control of the platform. It has been substituted later with the mobile custom applications.

Chapter 7

Architecture Design

It has been researched an effective solution for the problem of remotely control the system, to be more specific, by the use of a smartphone or tablet, adding, hence, an external component in order to mimic as much as possible the normal use that would be done with a classic television.

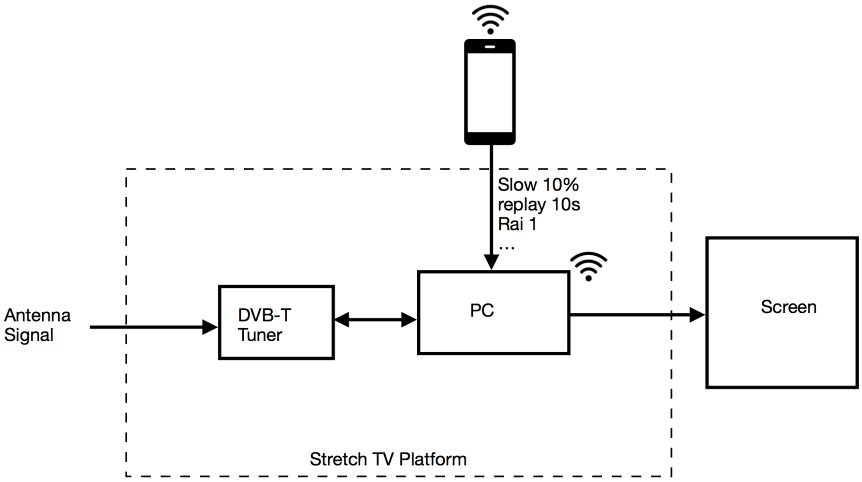


Figure 7.1. Stretch TV Platform Interaction With Remote Controller

We decided to offer the possibility to the final user to control the system, hence the capability of selecting among the various functionalities proposed by the Stretch TV Platform, by the use of a smartphone or tablet with a dedicated application. The various functionalities among which is possible to select are: the possibility to choose within different speed of reproduction of the multimedia streaming, from keeping the original speed to a reduction of 30% of it, the opportunity to choose among different type of audio equalizations that are going to emphasize more or less certain frequencies of the auditory

spectrum, the possibility of select among a vast selection of RAI television channels and the opportunity of performing an instant replay of the television content been watched.
In general, considering the whole architecture of the remote control system, it has to mimic the use of a normal television remote controller and it has to be as simple as possible in its use.
To make possible the interconnection and communication of the remote device with the Stretch TV platform, the whole infrastructure has been designed with the client server paradigm. Hence the server (the platform) is going to provide all the features of the system to the client (the smartphone or tablet) that connecting to the server will be capable of selecting among all the different options and features and finally the platform (server) will respond by going to perform all the operations needed to modify and manipulate the multimedia streaming for the specific selected functionality.

The necessity of find a clever way of performing software updates of the application running on the Stretch TV platform has been considered. This mainly because there is the need to update the server to correct error, improve performance or add new features even when the system in the near future will be sold.

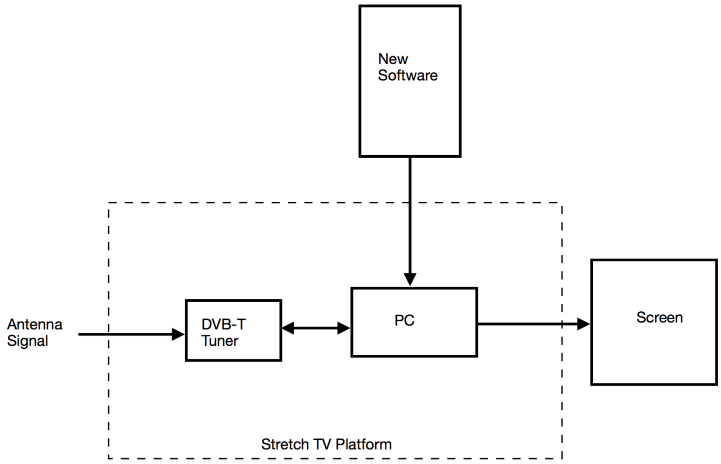


Figure 7.2. Stretch TV Platform Software Update

In this specific case, it has been chosen to implement this functionality in the remote controller application, avoiding hence the needs of a central server keeping tracks of all the Stretch TV platforms and providing updates when requested and necessary.

Chapter 8

Implementation

In this Thesis two custom mobile applications have been developed, one meant for the Apple iPad and the other meant for Android smartphones. With these application it is possible to connect with the Stretch TV platform and, through the user interface presented, control the operations that the system has to performs. The overall Stretch TV architecture has been designed and divided into two main components, one front-end, the mobile application, and one back-and the Stretch TV platform. Through the front-end it is possible to control the back-and by selecting which are the preferred configurations for the fruition of the television content and the platform will execute the corresponding operations. This architecture has been designed and developed in order to mimic the interaction with a normal remote controller allowing great flexibility and ease in the interaction with the platform. Moreover general improvements in the usability of the platform, correction of errors that causes critical failures of the system and enhancement in the multimedia processing processes management have been carried out to provide a more stable and reliable platform. First of all, and more important, there was the necessity of finding an efficient way for connecting the remote controller devices to the system platform. The principal problem in this case derives from the necessity of not knowing a priori the IP address of the machine connected in the local network. Mainly because in the future the Stretch TV platform will be used in people houses and to keep the use and installation of the platform easy and straightforward as much as possible.

8.1 Bonjour/Avahi Protocols

The Bonjour (available for Apple and Windows platform) or Avahi (available for Linux distributions)protocols are two implementations of the Zeroconf (Zero Configuration Networking) protocol, they allow to use the mDNS (multicast Domain Name System) service to resolve host names to IP addresses within small networks that do not include a local name server avoiding the needs of manual operator intervention or special configuration servers for configuring the network setting. The mDNS service allows to assign to a machine a unique name in the .local domain giving the possibility to other computers or devices, connected to the same network and that implements this service, to connect to

each other without the needs of knowing the exact IP address but just the name assigned on the .local domain.
It has to be considered that the mDNS service only resolves host names ending with the .local top-level domain hence it works only on local area network, but taking into account the needs of this Thesis works, and the Stretch TV platform in general, there is the need of accessing the platform within the local network and not from the outside.
Another interesting feature of the Bonjour/Avahi protocols is the possibility to register services published by devices connected on the local network and allowing to other devices that implements those protocols to search and automatically detect those services.

8.2 Server-Client Connection

Considering which are the devices that will be used for remotely control the Stretch TV platform, hence, Apple mobile devices using iOS, and any other Android devices, there are no problems in using the Bonjour protocol in Apple devices, since this protocol has been initially implemented by Apple itself, that natively support this features. The problem arises when considering Android devices; in fact Google override devices DNS entries forcing to use its own servers. This problem could be solved by having a rooted android device and manually edit some configuration files, moreover this approach require also that the Stretch TV has been assigned with a static IP address, but for the particular necessity of the Stretch TV project this is unfeasible in order to keep the installation and use of the platform as simple and straightforward as possible avoiding or reducing to the minimum the needs of manual intervention of the final user when installing and configuring the system.
In order to overcome this problem another feature of the Bonjour/Avahi protocol can come in handy, in fact this protocol allows also to register some devices services on the local network and allows to other devices, that implements Bonjour/Avahi protocol, to search and find those services. The service registration requires to assign a unique name for the service, it's type which contain the protocol used for communication, the domain in which register the service hence the .local domain, and the TCP or UDP port on which the service is listening. Given that in both Android and iOS applications a functionality for finding the specific web service provided by the Stretch TV platform has been implemented using native class, respectively NsdManager for Android and NetService for iOS; this requires to provide to those methods respectively the name used for register the service and the type. Then, once the service discovery has been resolved, the devices applications can have access to all the information required to connect to that service in particular the IP address of the machine publishing that service and the port on which the service is listening.
This methods of implementation and interaction with the system provides a great flexibility in the connection with the Stretch TV platform, avoiding the needs of knowing any particular information about the machine running the server application and the network configuration, only the informations about the registered service are required. It avoid also to the final user any network configuration procedures when installing the platform or the Android application. Moreover it also overcome any future problem deriving from

some changes in the local network configuration, or dynamic IP assignment for the Stretch TV platform. This also strongly simplify the platform installation in the final users homes avoiding any required specific network configuration.

8.3 Server-Client Communication And Graphic Interface

Moving on with the implementation phase, a communication protocol and interface have to be developed.

For doing this on the platform a software implementing the server is running, accepting any connection with outside devices. The server part is written using python, and in the specific leveraging on the Cherrypy minimalist web framework.

Cherrypy allows for an easy and robust way for developers to build web application in much the same way they would build any other object-oriented Python program, it is HTTP/1.1 compliant and can be used in different context and web server development.

In the specific context of the Stretch TV project it is used for providing a web server running on the Stretch TV platform providing access and services to remote devices responding in an adequate way to client requests, thus executing all the FFmpeg and tzap command required for the specific operation to be performed.

For the communication protocol it features WebSocket. WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection, it is a different TCP protocol with respect to HTTP but it is designed to work over HTTP thus making compatible with it. To achieve compatibility, hence, the WebSocket handshake uses the HTTP Upgrade Header to change from the HTTP protocol to the WebSocket protocol.

As stated before, unlike HTTP, WebSocket provides full-duplex communication between server and client over a single TCP connection, moreover it also allows for message to be passed back and forth between server and client while keeping the connection open making possible to reduce TCP handshake and HTTP header overhead low, thus making it efficient for small message exchange. The Cherrypy server was used initially to publish and interact with the HTML page used for controlling the system and later on has been integrated with the remote controller application to establish a connection and responding to message received in order to provide the correct functionalities to the system.

Going back to the actual remote controller application implementation, once the client has found the server information required for the connection, it moves on establishing a WebSocket connection, this is achieved using the Starscream framework in iOS, while for Android the native class WebSocket has been used.

So, after these two step for the connection phase, the communication with the server has been established and the application can send strings of messages to the server in order to notify which are the actions to be performed, while the communication channel is kept alive util the client disconnect from the server avoiding the need of establish new connection and making the message exchange phase simple and clean.

The messages exchange protocol is string based, the client will send the appropriate string based message representing the corresponding action to be performed while the server waits for message receiving and will perform the appropriate multimedia streaming manipulation operation, issuing the correct commands.

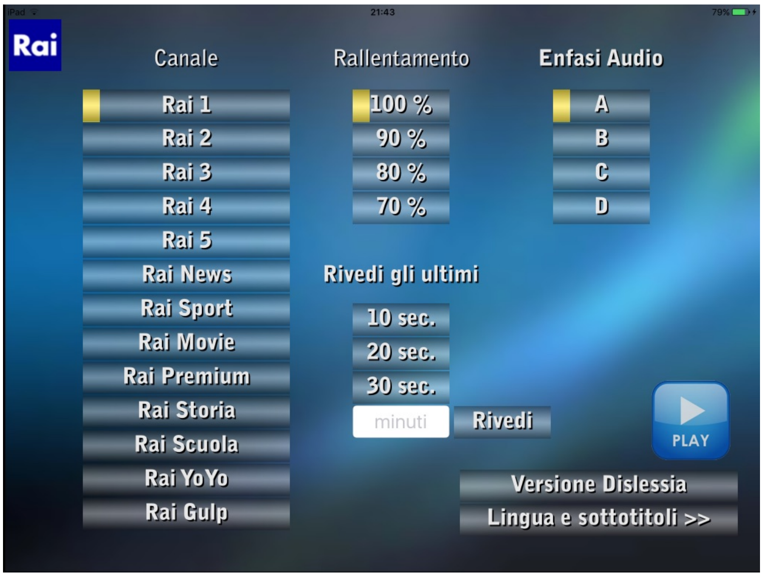


Figure 8.1. iOS version of Stretch TV application running on iPad

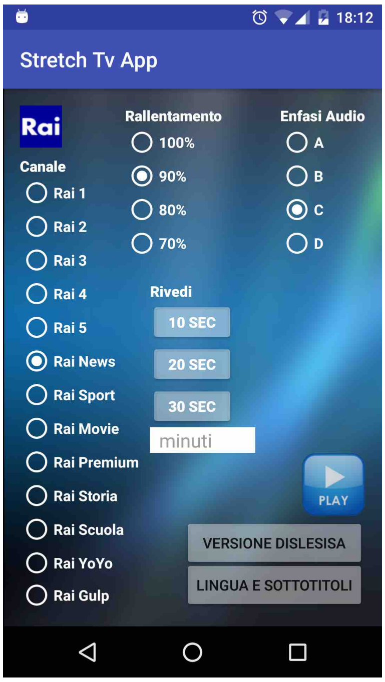


Figure 8.2. Android version of Stretch TV application running on Android smartphone

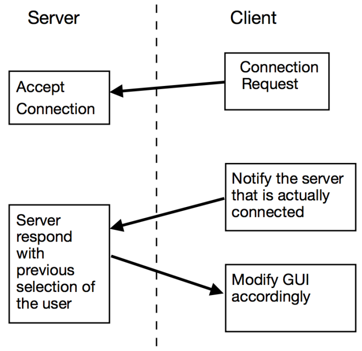

Server
Client
Accept Connection
Connection Request
Notify the server that is actually connected
Server respond with previous selection of the user
Modify GUI accordingly

Figure 8.3. Diagram of Message Exchange upon Connection

In Figure 8.1 and in Figure 8.2 are presented both version of the Stretch Tv project application, the graphic user interface design is based, concerning the iPad version. On the one explicitly requested by " RAI - Centro Ricerche e Innovazione Tecnologica, Unità Tecnologie per Accessibilità Prodotto RAI e Tecnologie per la e-Inclusion " while for the Android smartphone version is an adaptation of it, and contains all the features currently available on the Stretch TV platform.

The user can select among the different features proposed by the application interface, such as the channel to be shown, the speed of the reproduction of the multimedia streaming content and also the preferred audio equalization method to be adopted, than when the user hit the play button, the application provides to send all the useful information about the user choices to the server by means of predefined strings of messages and the server will provides to execute all the needed operations to make those choices effective. The replay functionalities are kept apart from the play button, and are implemented as an instant feature, hence when the user request for a replay it simply touch on the corresponding button an the application will send the corresponding message to the server. There are several options for the length of the time interval to replay such as 10, 20 and 30 seconds as well as the possibility to request a replay in the form of minutes of the television streaming to be shown again.

Moreover, an additional messages exchange phase has been added after the connection has been established, in this way the server can provide some information to the client upon each connection, in particular the server will keep tracks of the selection performed by the user and which are the current streaming characteristics, hence the channel currently shown, the speed of reproduction and the audio equalization being adopted.

Figure 8.3 represent an example diagram of a messages exchange between server and client upon connection. In this way, when the user launchs the remote controller application it can see which are the characteristics of the multimedia streaming currently been shown, if any, in order to keep tracks of which are the modification done before.

The initial message from the client was required in other to supply to a particular operating principle of Cherrypy, used for the server implementation. When connecting with the Cherrypy server, a method is called that allows to perform some preliminary configuration, but since the HTTP handshake required for the connection is still in execution it is not possible to send any messages to the client from the server, otherwise those messages could interfere with the HTTP handshake causing a failure in the client-server connection. Hence when the client application successfully connects with the server, it sends a message notifying that the connection has been established and the server can then start sending messages to the client safely. The possibility of keeping track of previous decision taken has been added to make more fluid and natural the user experience while interacting with the Stretch TV platform.

8.4 Platform Software Update

Another part of the development process was to find an effective and clear way to perform updates for the software running on the Stretch TV platform. It has been decided to implement this operation as a built-in option of the Stretch TV mobile application to avoid the needs of keeping the Stretch TV platform connected to the net and having a central server managing the update process of each machine. This implementation requires to add a management layer to the server, in particular the possibility to receive the new version of the software and perform a restart of the service, as well as an additional messages exchange phase on the client-server connection startup.

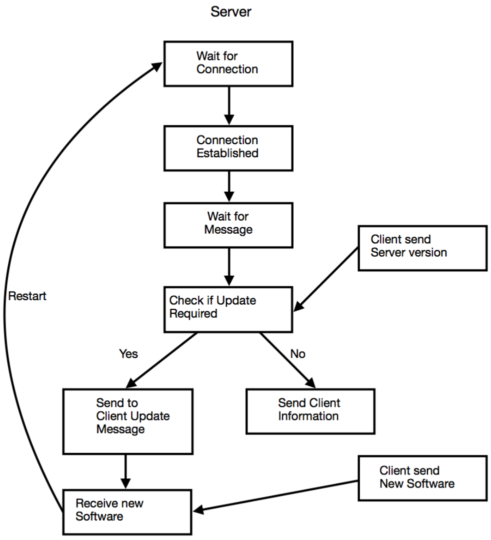


Figure 8.4. Diagram of Message Exchange upon Connection with Update

Once the client is successfully connected with the server it send a message containing the release version of the server required by the application, then the server upon receiving the message checks wether its own current version is the same as the one required by the client application or not. In case not, it sends a message to the client requesting an update. Once the client receive the update message from the server, it shows to the user a new screen informing the user that an update is required and than the user can begin the update.

To speed up the connection, the release version message substitutes the message sent by the client to notify the server that a connection has been established and the server can send messages to the client as shown in Figure 8.4.

When the user starts the update, it is sent to the server the new version of the software, and the server provides to save it and replace the older version with the new one. Once this operation has been done, the server is automatically restarted loading the new version of the software. Finally the user can reconnect the remote device to the Stretch TV platform gaining access to the new features added or taking advantage off all the changes being done.

Once the server sends the update request, it is not blocked on waiting for the new version, hence if the user for some reason, before issuing the update, uses an application that still works properly on the current installed release version of the server software, it can interacts with it without any particular problems, also without the need of restarting the machine.

Moreover, the choice of implementing the server update through the application, can make the user experience more fluid and easy, in fact util the user decides to update the application, it can interacts with the server without causing any problems, avoiding the possibility of some errors or limitations in the use of the application due to a previous server update and not of the remote controller application. Furthermore, the server update is not bind only for newer versions, but, potentially, if the user uses an earlier version of the application with respect to the one currently running on the Stretch TV platform, the server will request an update (in this case a downgrade will be executed to be more precise) in order to keep the server functionality aligned with the ones proposed by the remote controller application.

8.5 User Interface And System Enhancements

A last addition to the graphic user interface of the application has been carried out, it gives the possibility to the user to change the font of the application to one expressly designed to help in the reading process people affected by dyslexia.

The font selected is the Biancoenero[1], it is the first Italian font with high readability capabilities. Each character design has been analyzed and changed in a way that it is not possible to mistake, especially in case of specular ones such as **b-d**, **p-q**, **a-e**. The characters that in common font presents similarities, such as **l** and **I** or **m** with **n** and many other, have been redesigned in order to make them more differentiated. Furthermore the average width of characters has been increased with respect to normal fonts, the kerning and the space between words have been also increased proportionally.

[1] http://www.biancoeneroedizioni.it/font/

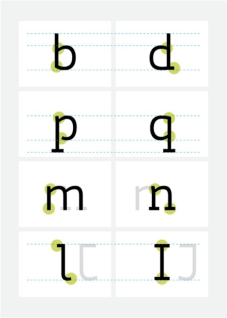
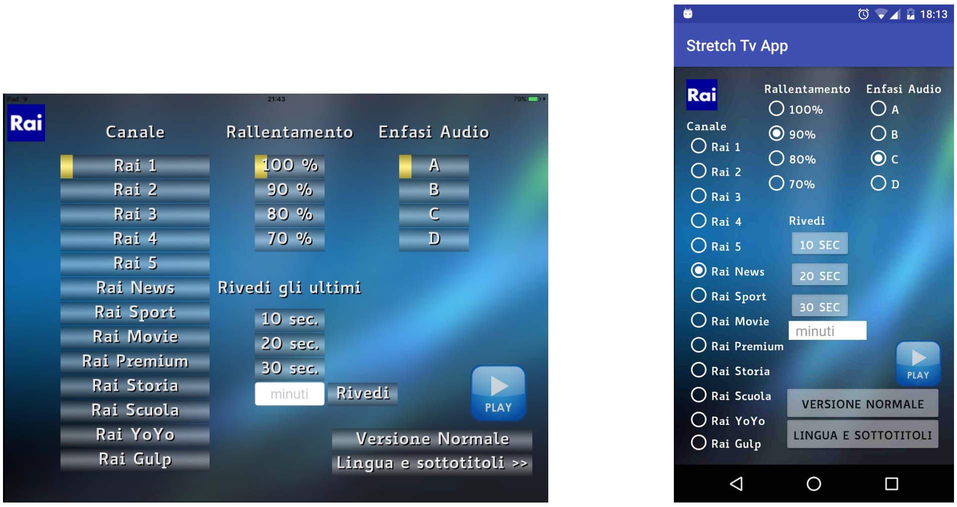


Figure 8.5. Example of Biancoenero Font Character Design

The interest of the Biancoenero font project is to introduce some formal variation, functional to readers affected by dyslexia without penalizing the overall typographical quality [2].



Canale: ○ Rai 1 ○ Rai 2 ○ Rai 3 ○ Rai 4 ○ Rai 5 ● Rai News ○ Rai Sport ○ Rai Movie ○ Rai Premium ○ Rai Storia ○ Rai Scuola ○ Rai YoYo ○ Rai Gulp

Rallentamento: ○ 100% ● 90% ○ 80% ○ 70%

Enfasi Audio: ○ A ○ B ● C ○ D

Figure 8.6. Application Graphics Interface with Biancoenero Font

As a final note, the remote controller application also features the possibility to access a new screen that will provide the user the capabilities to select among different option for languages and subtitles. Even if this features is already built in the application, it has not yet implemented on the Stretch TV platform and has been added only for demonstration purposes.

Finally the remote controller application has been integrated with the existent platform, adding al the new features required, such as more channels among which selects or

[2] Luciano Perondi, Biancoenero font designer

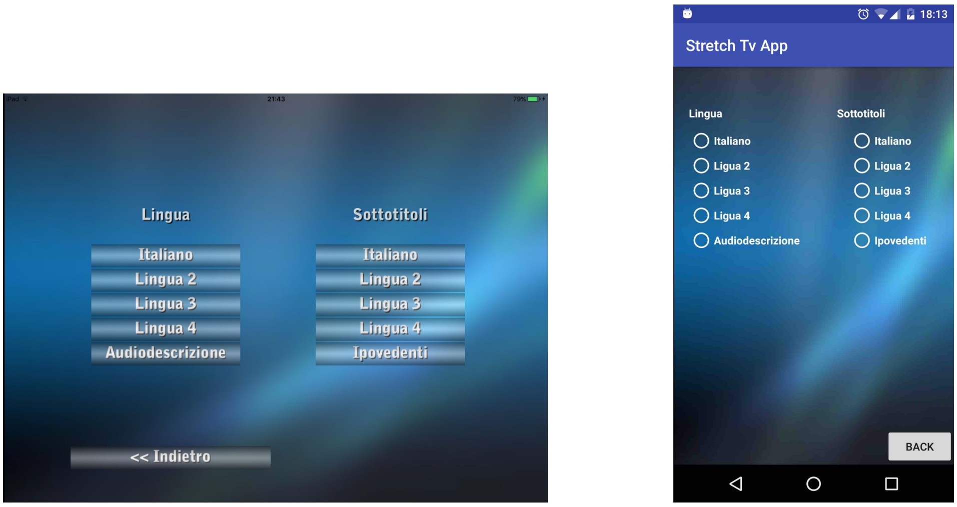


Figure 8.7. Immage of Language and Subtitle Application Page

more options for the replay features.
Moreover improvements on the usability of the platform have been carried out. Errors that causes critical failure of the system have been corrected and improvements in the management of multimedia processing processes have been developed to provide a more stable and reliable platform.

Overall all this additions to the Stretch TV platform, and the integration of a remote controller capability to the project, have added an easy and straightforward way to interact with the system for personalizing the fruition of the television content and making, hence, a great step in the direction of a possible use by future customers.

Chapter 9

Results And Discussion

After the integration of the Stretch TV platform and the work carried out during this Thesis, performance tests have been performed to better understand which are the strength and weakness of the current approach, in particular to demonstrate that the actual platform is not capable of managing high definition contents without loosing in quality or intelligibility of the output video streaming, hence provide a starting point for possible future developments and improvements.

9.1 Focus Group Stretch Tv

Before moving on with the results of the performance tests done, is interesting to have a look at some of the result of a Focus Group[1] performed by " RAI - Centro Ricerche e Innovazione Tecnologica, Unità Tecnologie per Accessibilità Prodotto RAI e Tecnologie per la e-Inclusion " to better understand how the final users react to the Stretch TV project and all the possible improvements, on the fruition of the television contents, it can bring to people.

The Focus Group consisted of proposing to people a questionnarie with several general questions such as the age of participants, their habits in using television or other services made available by RAI, such as the RAI Play video on-demand platform, moreover there has been proposed to people also several excerpt of television news. These multimedia contents have been differentiated based on the nature of the news transmitted and two main division have been made: TV news containing news with low emotional impact, such as normal report or political news, while the other category was TV news with high emotional impact contents, such as crime news or war reports.

There were analyzed thirty different television news video and from each of them there were carefully extracted different videos in order to cover all the possible different type of contents.

For each of the different TV news contents shown there have been proposed also different

[1] http://einclusionprojects-crit.rai.it/StretchTV_Web_Monitoring/Grafici_finale_mobile/

reproduction speeds, ranging from the original speed to a reduction of 20% of the multimedia contents. The speed reduction percentages have been selected by " RAI - Centro Ricerche e Innovazione Tecnologica, Unità Tecnologie per Accessibilità Prodotto RAI e Tecnologie per la e-Inclusion " based on the typology of the news content proposed and on the prosody of the speaker, hence, based on some listening test effectuated, an optimal range of slowing down between 80 and 90% has been selected.

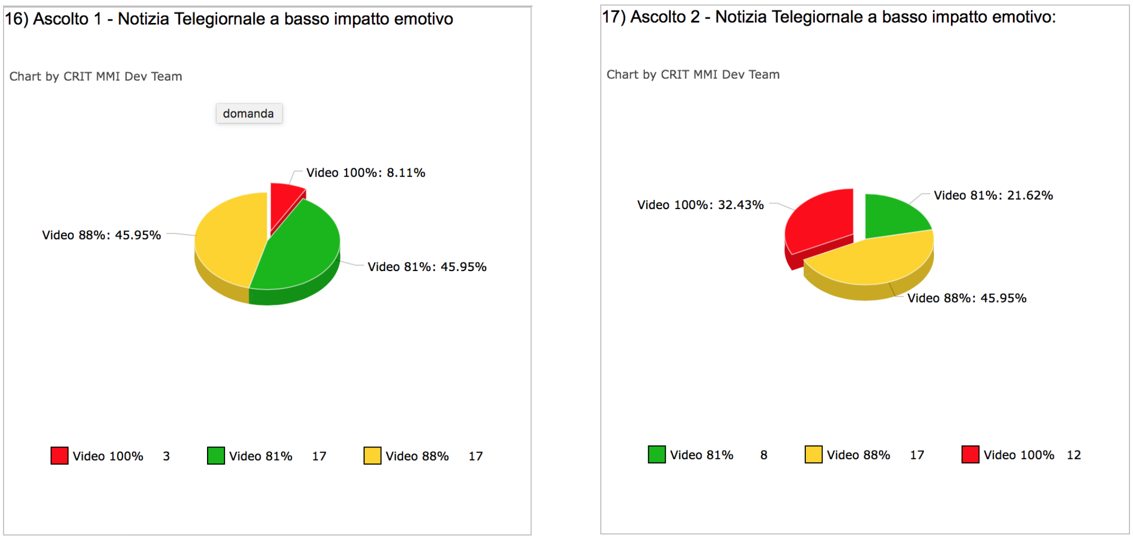


Figure 9.1. First and Second Listening, TV news Low Emotional Impact

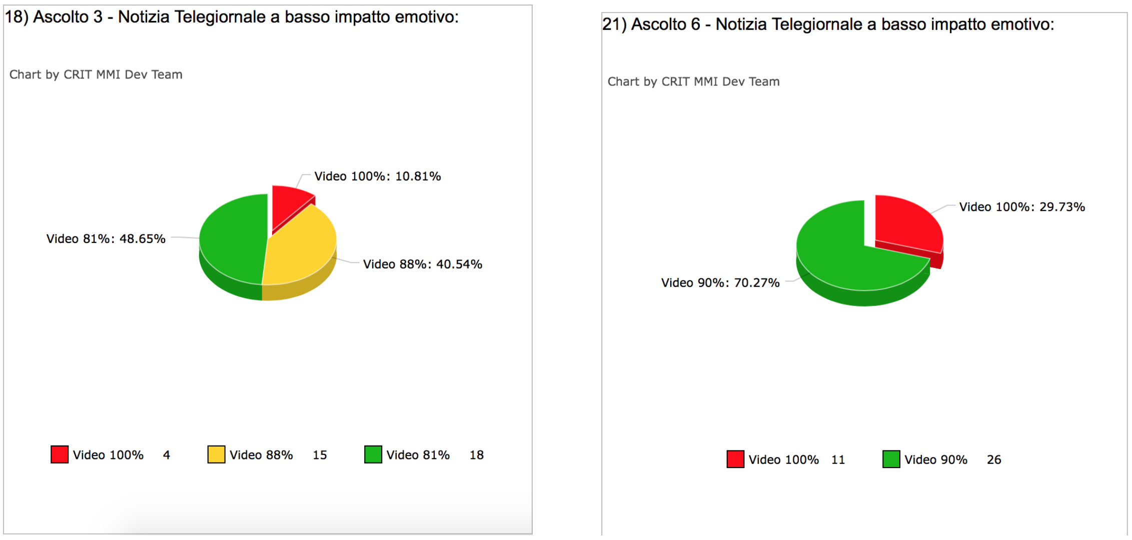


Figure 9.2. Third and Sixth Listening, TV news Low Emotional Impact

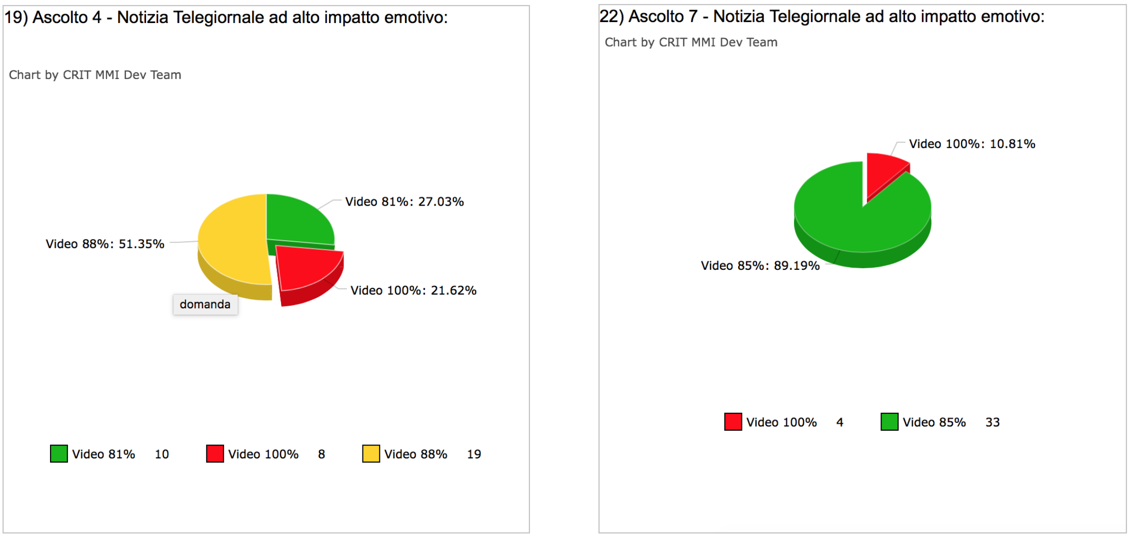


Figure 9.3. Fourth and Seventh Listening, TV news Low Emotional Impact

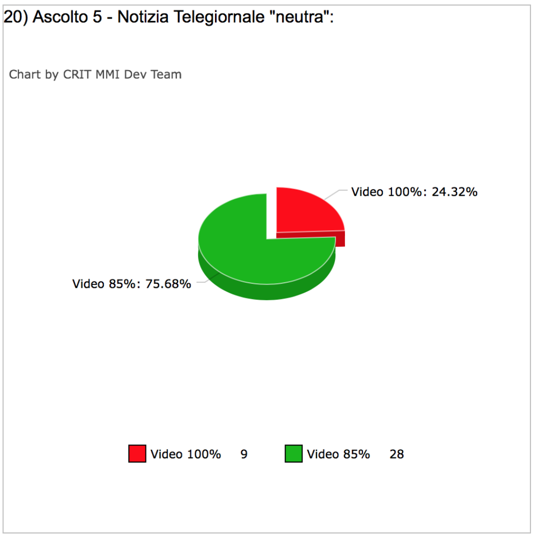


Figure 9.4. Fifth Listening, TV news "Neutral" Emotional Impact

Figures 9.1, 9.2, 9.3, 9.4 report how the various people involved in Focus Group have perceived the audiovisual contents proposed, and in particular which were the preferred percentages of slowing down for a comfortable listening. Considering the age range of the people targeted in this Focus Group, it starts from 51 years old with the majority of people over 70. It has been noticed that a large majority of the people involved in this Focus Group, no matter what were the contents of the TV news transmitted, preferred the fruition of a slowed down version of the television contents for a more comfortable

listening, while the percentage of the better listening experience may vary depending on the speech speed of the speaker.
Moreover, based on the result of the Focus Group, has been decided to select three different slowing down options to be integrated in the Stretch TV platform, 10, 20 and 30%, as well as the option to maintain the original multimedia streaming reproduction speed.

9.2 Performance Tests

Performance tests have been carried out in order to better understand which are the actual capabilities of the platform and have been performed by analyzing the time required by the platform to manipulate the multimedia streaming as well as which are the possible impact on the overall platform performance of the various processing steps.
As a clarification, for better understand the various tests performed, the functionalities of FFmpeg are based on the concepts of filters. Filters are the various manipulations that can be applied to the multimedia streaming and different filters can be chained together in order to performs different operations. The filters used for these tests are:

- **yadif**: filter used for deinterlacing the video. The deinterlacing filter is used to obtain a better quality video on the output, and in the particular context of the operation performed is used for obtaining from one frame of the input streaming one frame of the output streaming.
- **atempo**: filter used for slowing down the audio.
- **setpts**: filter for slowing down video.

It is important to clarify that all test were performed under the same system condition in order to obtain as much as possible similar result on the CUP usage parameter for a better comparison of the final results. For the CPU Usage percentage, since two cores are available to the system, the CPU usage can range from 0 up to a maximum of 200%.

The first test carried out involved the use of two different source video, one in Standard Definition and one in High Definition, both of the same fixed length, e.g. one minute. The test consisted in analyzing the system performance when no streaming manipulations were involved. The source video was stored in memory, through the FFmpeg command, it has been decoded, encoded and written back in memory without the use of any additional filters.

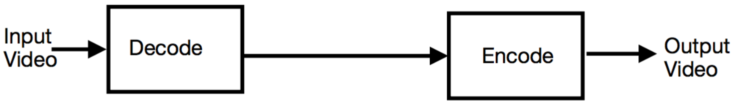


Figure 9.5. Operation Performed In The Test

Video Definition	CPU Usage	Memory Occupation	Time Required
	%	MB	sec
SD	135.09	53.24	5.948
HD	195.45	544.768	164.472

Table 9.1. HD and SD videos, no operation performed, video length 1.01 minutes.

The results obtained, shown in Table 9.1, clearly demonstrate that, while for the Standard Definition video there was no problem for the system in performing these operations, for the High Definition one instead, the system requires almost three time of the video length, in terms of time required, for performing these operation using almost all the CPU computation capabilities.

Another test performed involved the use of a Standard Definition video of fixed length, e.g., one minute.

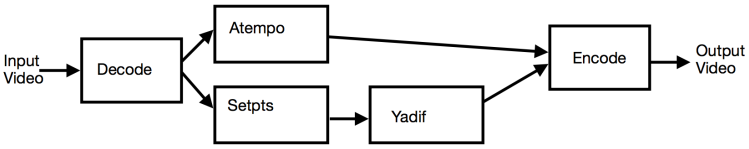


Figure 9.6. Operation Performed In The Test

Slow Down Applied	CPU Usage	Memory Occupation	Time Required
%	%	MB	sec
0	120.57	53.24	8.772
10	120.57	53.24	8.836
20	120.86	53.24	8.902
30	121.18	53.24	9.244

Table 9.2. Standard Definition video, all filters used, video length 1.01 minutes

Table 9.2 reports the result of the tests for each of the different slowing down options integrated in the Stretch TV platform. With respect to the results shown in Table 9.1, in this case the time required was a little bit worse, it is also required taking into account that the percentage of CPU usage in this case is a little bit less, while the tests performed involved the use of additional streaming manipulations that slowed down the overall execution. Moreover, a slight difference in the time required comparing the different slowing down options can be noticed, this difference is mainly due to the time required for storing the output video, that, in case of a slowed down version, is longer with respect to the original, and also for the greater use of the deinterlacing filter that, for a slowed down version of the video, requires more works due to the increased number of video frames.

A similar test case scenario involved the use of a High Definition video of fixed length, e.g., one minute.

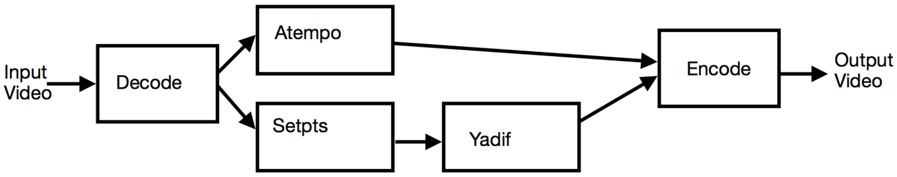


Figure 9.7. Operation Performed In The Test

Slow Down Applied	CPU Usage	Memory Occupation	Time Required
%	%	MB	sec
0	195.62	544.768	179.454
10	195.54	548.864	189.987
20	195.34	548.864	190.270
30	195.52	548.864	194.167

Table 9.3. High Definition video, all filters used, video length 1.01 minutes

Table 9.3 reports the results of the tests performed. In this case the result do not differs from what can be expected looking at previous test cases. Also in this context the differences in performance for the various slowing down options analyzed were due to the increasing length of the output video and the more works required by the use of the deinterlacing filter. It is important to notice that in this case, where all filters required for the correct functioning of the Stretch TV platform were used, the overall time required for manipulating a High Definition video was almost three times the original video length or even slightly more, hence making impossible for the platform to works on real time multimedia streaming manipulation of High Definition videos.

Another series of test has been performed in order to better understand which is the impact on performance of the various filters.
The videos used are respectively in Standard and High definition, same fixed length.

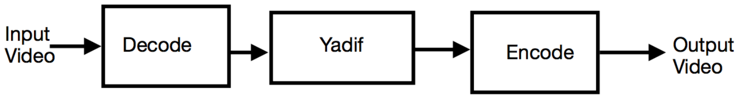


Figure 9.8. Operation Performed In The Test

Video Definition	CPU Usage	Memory Occupation	Time Required
	%	MB	sec
SD	127.48	53.24	8.205
HD	195.04	548.864	174.248

Table 9.4. HD and SD videos, only yadif filter, video length 1.01 minutes.

Table 9.4 reports the result of the tests. As can be noticed with respect to the results obtained in Table 9.1 where no streaming manipulations operations were involved, there is a noticeable increase of the time required, especially in the case of High Definition video where the CPU usage is almost the same as before. These tests reveal that the deintelacing filter has a noticeable impact on the overall performance of the video manipulation operation, but this filter has been necessary in order to obtain a video output with a good quality and it can not be removed.

Furthermore, another series of tests has been carried out to understand the impact on performance of the slowing down operation. The test conditions remain as before, one video in Standard Definition and one video in High Definition both of the same fixed lenght.

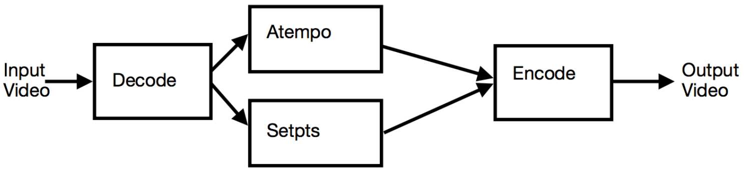


Figure 9.9. Operation Performed In The Test

Slow Down Applied	CPU Usage	Memory Occupation	Time Required
%	%	MB	sec
0	131.80	53.24	6.543
10	131.00	53.24	6.473
20	131.42	53.24	6.598
30	131.22	53.24	6.660

Table 9.5. Standard Definition video, only slowing down filters used, video length 1.01 minutes

Table 9.5 reports the results of the tests applied on the Standard Definition video. It can be noticed that the slowing down part, especially by looking at the fist row of the table, do not impact to much on the overall performance. While if the operation of storing the video is also considered, it is possible to notice a progressive decrease in performance, more time required for executing the FFmpeg command, mainly due to the need of storing a larger video on the output.

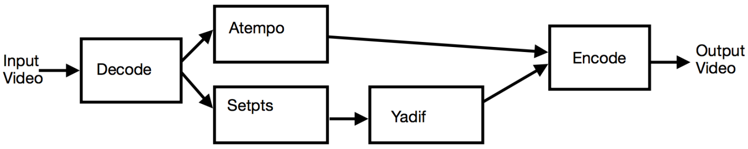


Figure 9.10. Operation Performed In The Test

Slow Down Applied	CPU Usage	Memory Occupation	Time Required
%	%	MB	sec
0	195.46	540.512	166.856
10	195.23	540.512	169.313
20	194.66	540.672	177.994
30	195.17	540.672	186.181

Table 9.6. High Definition video, only slowing down filters used, video length 1.01 minutes

Table 9.6 reports the results of the tests applied on the High Definition video. Also in this case the same consideration as the case of Standard Definition video can be done. It can be noticed that the slowing down filters did not have a large impact on the overall performance, the main difference is that while the slowing down percentage increase also the output video dimension increase, hence more time for storing the output video in memory is required.

The last performance tests carried out involved the measurements on how much time is required from when the FFmpeg command is issued to the start of the streaming output.

In order to show the video the ffplay tool is used. It is a tool include in the FFmpeg multimedia framework and it behaves as a video player.
For performing this tests two different video are used one in High Definition and one in Standard Definition with all filters applied.

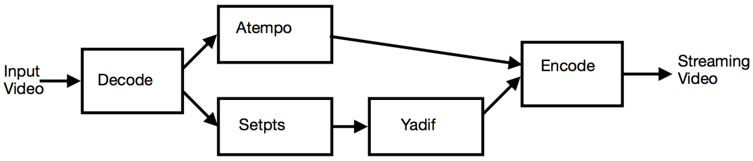


Figure 9.11. Operation Performed In The Test

Video Definition	Time Delay
	sec
SD	0.239
HD	2.218

Table 9.7. HD and SD videos, ffplay start delay.

Table 9.7 reports the results of the tests involving the measurement in the delay from when the FFmpeg command is issued to when the ffplay tool starts displaying the output video.
Also in this case a difference in performance between using a Standard Definition video and a High Definition video can be noticed, this difference is due to the more time required to perform all the operation before starting the video streaming.

9.3 Qualitative Tests

Overall all the streaming manipulation operation does not impact on the output video quality.

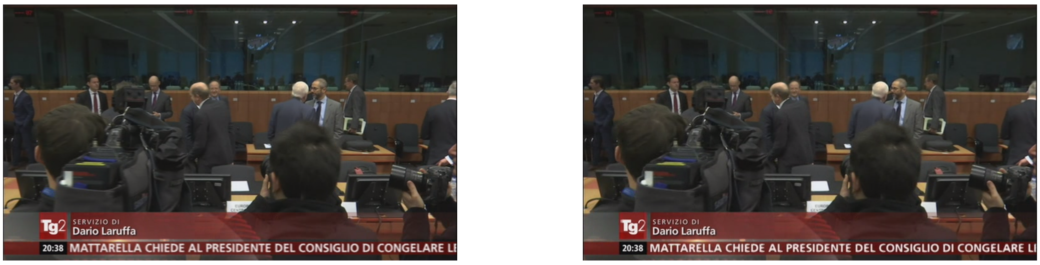

Tg2
SERVIZIO DI
Dario Laruffa
20:38 MATTARELLA CHIEDE AL PRESIDENTE DEL CONSIGLIO DI CONGELARE LE
Tg2
SERVIZIO DI
Dario Laruffa
20:38 MATTARELLA CHIEDE AL PRESIDENTE DEL CONSIGLIO DI CONGELARE LI

Figure 9.12. Standard Definition video before and after slowing down and audio equalization applied

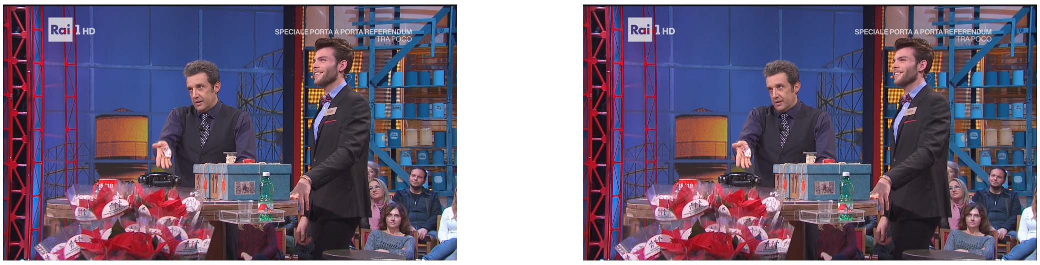

Rai 1 HD
SPECIALE PORTA A PORTA REFERENDUM
Rai 1 HD
SPECIALE PORTA A PORTA REFERENDUM

Figure 9.13. High Definition video before and after slowing down and audio equalization applied

Figure 9.12 and in Figure 9.13 show that, there is no noticeable difference in quality from the source to the output result after the manipulation.The audio equalization or slowing down operations does not impact on the intelligibility of the final television content. Hence making the use of the Stretch TV platform comfortable, in the fruition of television content, with the same video quality as the original but with the possibility of an improved experience in the consumption of the multimedia streaming in terms of intelligibility and comprehension.

Considering all the tests and results shown above, it can be noticed how, actually, the platform has no problem in the manipulation of the television content in Standard Definition, while for High Definition the actual Stretch TV platform does not have enough computing capabilities for managing that type of content while producing a seamless output video, in fact the fruition of the High Definition content is impossible due to a lagging streaming output that strongly reduce the intelligibility and comprehension of the multimedia content. It has to be considered that the Stretch TV platform software runs on a virtual machine and this has a huge impact on the overall performance since only half of the hardware resources of the computer are made available to the platform. Moreover, it has to be taken into account that, all the more heavy tasks of the multimedia manipulation process, in particular the video transcoding phase, are done in software without the help of any hardware acceleration technique. Surely the possibility of taking advantage of a GPU for the video transcoding phase can come in handy, greatly improving performance and speeding up the overall process, giving the possibility to use higher definition contents. In particular, in this case, it has not been possible to exploit such possibility due to the fact that the software runs on a virtual machine and the computer does not have a dedicated GPU, hence was impossible to take advantage of the integrated graphic card.

Chapter 10

Conclusion and Future Works

Summarizing, the Stretch TV project is a huge project. Its main aims is to bring enhancement in the fruition of the television contents especially to elders and people affected by sensorial and/or cognitive disability. Moreover, it has also the potentiality to be used by foreigners that are learning Italian, or more in general by people that have or want to learn a new language. In fact the fruition of a slowed down television content can improve and ease the listening and comprehension of the multimedia content. Furthermore, also audio equalization techniques have been considered in order to guarantee an optimal listening, also to those people that are affected by a partial hearing loss, in particular elders that suffers from this problem as a normal agin process of the auditory system, avoiding hence the needs of increasing the television volumes, but instead by emphasizing those frequencies of the auditory spectrum on which the hearing is laking.
During this Thesis work, a custom mobile application, for remotely control the system, has been integrated with the Stretch TV platform, bringing the interaction with the platform even more towards which could be a natural use of a normal television, hence making the use use the Stretch TV platform more easy and flexible. During the development and integration of the remote control system, particular attention has been made in reducing as much as possible problem that could arise due to an erroneous installation of the platform in people houses, trying hence to avoid any required manual intervention in the configuration, in particular the need of a configuration process of the local network, moving towards the concepts of a plug-and-play system.
Moreover, improvements in the usability of the mobile application have been made, trying to make the user interface as simple as possible.

Another aim of the Stretch TV project is to develop a cheap platform with the required computing capabilities for manipulation several type of multimedia contents even with higher video definitions without degrading the overall quality of the audiovisual streaming. In this, the actual platform is not compliant, but it has to be considered that it was developed in a first place for demonstration purpose and not with the aim of been produced for people. As a result, it has been developed a first prototype integrating all the functionalities required, but with form factor and price not adequate. For possible future development, the Apple Mac Mini will have to be substituted by an ad-hoc device,

cheap and that supports hardware acceleration. An example could be the investigation of possible platforms based on the vast variety of Android smart Tv boxes that take advantage of the Rockchip dedicated graphic component, that allows to keep the price low while guarantee enough computing capabilities for manipulating higher definitions streaming.
In general, there is room for a variety of possible future modifications and integrations to the Stretch TV platform, such as the addition of different language and subtitles for the fruition of the television contents, that is already integrated in the remote controller application, but not in the platform itself and will be surely object of future studies.
A study on how people react to the use of the mobile applications and their graphic interfaces could be a good starting point for possible improvements and simplification of the remote controller to further enhance the user experience in the interaction with the platform. Moreover, tanks to the use of a custom application for the remote control and interaction with the Stretch TV platform, it could be possible to investigate the possibility of developing a more modular system, where the final user could decide which are the features that it prefers more and on which is more interested, removing those that are not necessary, making the use of the custom application and the system more agile and focused on the users needs.

Bibliography

[1] Studio Di Otorinolaringoiatria Tanzariello, *Presbiacusia*, 2016. Available at: http://www.tanzariello.it/index.php/orecchio/patologia-orecchio/orecchio-medio/145-studio-prof-a-tanzariello/orecchio/patologia/orecchio-interno/ipoacusia-neurosensoriale-dell-adulto-di-tipo-progressivo/781-presbiacusia. Accessed 6 November 2017.

[2] A. Hayati, *The Effect of Speech Rate on Listening Comprehension of EFL Learners*, 2010, pp. 107-114.

[3] Y. Zhao, *The Effects of Listeners' Control of Speech Rate on Second Language Comprehension*, in "Applied Linguistics", Vol. 18, No 1, 1997, pp. 49-68.

[4] E. Buroni, *La voce del telegiornale. Aspetti prosodici del parlato telegiornalistico italiano in chiave diacrona*, in “L'italiano televisivo 1976-2006, Atti del Convegno Milano, 15-16 giugno 2009”, 2010, pp. 387-408.