Universidade do Minho
Escola de Engenharia

Zacarias Benta

Leveraging Cloud Computing for IPTV: Moving the Set-Top Box to the Cloud

Abril de 2013

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Zacarias Benta

Leveraging Cloud Computing for IPTV: Moving the Set-Top Box to the Cloud

Dissertação de Mestrado
Mestrado em Redes de Comunicações e Serviços

Trabalho realizado sob orientação de

Professor Joaquim Macedo

Abril de 2013

Dedicated to my Grandmother Maria.

Imagination

`Imagination is more important than knowledge.`
`For knowledge is limited to all we now know and understand,`
`while imagination embraces the entire world,`
`and all there ever will be to know and understand.`

Albert Einstein

Thanks To...

I would like to take the time to thank my travel companion Professor Joaquim Macedo for his support and guidance across this adventure. I would also like to thank my family for all their support and comprehension, to Armin Van Buuren for his music that kept me going though the long nights of study and last but no the least to my lovely wife Susana for always believing in me and making me believe in myself.

Leveraging Cloud Computing for IPTV:Moving Set-Top Box to the Cloud

Abstract

IP Television (IPTV) has changed the way we perceive our TV sets. It has given us the ability to take control of our TV viewing activities.
We can now interact with our TV set in a way we might have thought impossible a few decades ago.
With the push of a button we can schedule TV recordings, stop and even rewind the broadcast. Users are now able to control what they want to view, when they want to view it.

TV sets have also evolved and gained a new set of functionalities, they have become smart, allowing us to connect o the Internet and decode media files that are inputed through usb ports and streamed through the network or the Internet.

But the TV is no longer the center of our multimedia experience, now we have new ways to access the TV content we have subscribed. Viewing TV on a laptop, a smartphone or even a tablet is now more and more common.

Cloud Computing (CC) has also brought us some revolutions, we can now have systems that grow and adapt on-the-fly to the conditions presented to them.
An application can extend it's storage capacity in a matter of minutes, and the same can be said of the processing power of the underlying platform.
Our objective in this work is to discuss the possibility of creating a synergy between both services.
By creating a system where IPTV and CC would interact, we could create a service that would provide IPTV anytime and anywhere.

With the possibilities CC brings we can move the Set-top Box (STB) to the cloud and create new service functionalities and reduce the cost of having to install a STB in every client.

We have focussed our work in the simulation of a CC infrastructure that would host Virtual STB's that would be accessible from wherever there would be a network connection.

Keywords: IPTV, CloudComputing, Virtual STB

Potenciação da Computação em Cloud Computing para IPTV: Colocação da Set-Top Box na Cloud

Resumo

O IPTV mudou a maneira como vemos os nossos aparelhos de TV, esta tecnologia permitiu-nos assumir o controle das nossas actividades de visualização de TV. Agora podemos interagir com a nossa TV de uma maneira que se pensaria ser impossível há algumas décadas atrás.

Com o toque de um botão é possível agendar gravações de TV, parar e até mesmo voltar atrás na emissão. Actualmente, os utilizadores são capazes de controlar o que querem ver e quando o querem ver.

As TVs também têm evoluído e ganharam um novo conjunto de funções / ferramentas, tornaram-se inteligentes, o que lhes permitiu efectuar ligações á Internet e descodificar ficheiros multimédia que são introduzidos através das portas USB que também ganharam. A TV deixou de ser o centro de atenções no que concerne ao nosso consumo de multimédia, agora temos novas maneiras de aceder aos conteúdos de TV que subscrevemos. Visualizar TV num portátil, smartphone ou até mesmo num tablet é cada vez mais comum.

O CC também nos proporcionou algumas evoluções, agora podemos ter sistemas que podem crescer e adaptar-se em tempo real, dependendo das necessidades. Uma aplicação pode aumentar o tamanho do seu sistema armazenamento numa questão de minutos, e o mesmo pode ser dito da capacidade de processamento da plataforma subjacente.

O objectivo deste trabalho é discutir a possibilidade de criar uma sinergia entre ambos os serviços. Ao criar um sistema onde o IPTV e o CC interagissem, poderíamos criar um serviço que poderia fornecer IPTV a qualquer hora e em qualquer lugar. Com o CC temos a possibilidade de passar a STB para a nuvem, criar novas funcionalidades, serviços e ao mesmo tempo reduzir o custo de ter que instalar uma STB em cada cliente. O foco do nosso trabalho foi na simulação de uma infra-estrutura CC que permitisse a criação de uma STB Virtual, podendo esta estar acessível a partir de qualquer lugar onde quer que existisse uma ligação à rede.

Palavras Chave: IPTV, CloudComputing, Virtual STB

Contents

List of Figures

List of Tables

Acronyms

3GPP 3rd Generation Partnership Project

API Application Programming Interface

ADSL Asynchronous Digital Subscriber Line

CC Cloud Computing

COPS Common Open Policy Service

CPU Central Processing Unit

DRM Digital Rights Management

DSLAM Digital Subscriber Line Access Multiplexer

DVB-T Digital Video Broadcast Terrestrial

DVR Digital Video Recorder

ETSI European Telecommunications Standards Institute

FG Focus Group

FTTB Fiber To The Building

FTTH Fiber To The Home

FTTN Fiber To The Node

FTTX Fiber To The X

GC Grid Computing

GUI Graphical User Interface

HD High Definition

HTML Hyper Text Markup Language

HTTP Hyper Text Transferences Protocol

IaaS Infrastructure as a Service

ICC Instant Channel Change

IGMP Internet Group Multicast Protocol

IMS IP Multimedia Subsystem

IP Internet Protocol

IPTV IP Television

ISP Internet Service Provider

IT Information Technology

ITU International Telecommunications Union

Mbps Megabit per second

NAL Network Abstraction Layer

NASS Network Attachment Subsystem

NGN Next Generation Network

OLT Optical Line Terminal

OVF Open Virtual Machine Format

PaaS Platform as a Service

PIP Picture In Picture

PSTN Public Switched Telephone Network

QoE Quality of Experience

QoS Quality of Service

RACS Resource Admission Control Subsystem

RTSP Real Time Streaming Protocol

S-STB Software Set Top Box

SaaS Software as a Service

SD Standard Definition

SIP Session Initiation Protocol

SMSR Stream Management Stream Request

STB Set-top Box

Telco Telecommunications Operator

TISPAN Telecommunications and Internet converged Services and Protocols for Advanced Networking

VO Virtual Organisation

VOD Video On Demand

VoIP Voice over IP

VPN Virtual Private Networks

XDSL Digital Subscriber Line

Chapter 1

Introduction

1.1 Introduction

Internet Protocol (IP) networks have evolved from being the solution for data communication between computers. They can now go farther and further providing greater reliability and Quality of Service (QoS). We are now witnessing the appearance of an "All-IP" network as the evolution of this protocol suite who has allowed for the convergence of major services(Data, Voice, TV ...). This "All-IP" network is a synonym of Next Generation Network (NGN)[14].

With the evolution of the IP protocol there has also been an rise of applications with more strict QoS requirements. Besides elastic applications, currently we are surrounded by a new set of diversified applications with special needs regarding bandwidth, packet loss, delay and jitter. Applications, with strict QoS requirements ,like Voice over IP (VoIP) and IPTV are now very common.

Infrastructure Evolution

We are evolving for that "All IP" network that we have mentioned previously, and to be able to provide it to the home/corporate users there is a need for new infrastructures. Although the old copper wires can still support the current services, in the long run there will be a need for greater bandwidth.

The access infra-structure has also evolved, nowadays we have new technologies like Digital Subscriber Line (XDSL) and Fiber To The X (FTTX) that support high speed connections and offer greater availability. The popularity and price drop for both Internet Service Provider (ISP) and users of these kind of technologies, have turned them prone to be used as residential and corporate means of access and distribution of services/content.

Users Evolution

The engine that turned the wheels of evolution, when it comes to networking technologies, is the user. Users have a great demand for quality, reliability and speed when it comes to networking, they tend to demand for more bandwidth, more uptime and less errors.

Users are also becoming more informed, they are aware of the capabilities and some of the technologies behind networks, and claim for new types of services.

Services like VoIP and IPTV are now more common then they where 5 years ago, ISP's are receiving more and more requests to have these kind of features available among their product offer. On the other hand, we also have ISP's, that plan and implement more aggressive commercial campaigns with the offer of products for every user's needs.

TV Evolution

The evolution of users has also brought an evolution of their demands regarding currently available media. The television set has been the center of our entertainment life for about 60 years, then we had a shift on users interests and the computer took over with the Internet and it's social networks and video sharing sites.

Now we are seeing that the users are no longer passive, their request for interactive media has brought the TV to the Internet, and the junction of both gave birth to IPTV. Users can now take control of their TV anytime and anywhere. As TV runs under IP it can also be viewed in computers and cellphones, users can pause, rewind, record and even schedule recordings of their favorite shows.

1.2 Motivation

Most ISP's offer the IPTV service as the solution to the lag between the user's everyday life and the TV broadcast. They offer the ability to schedule recordings and stop/rewind capabilities, but they miss the anytime/anywhere option that most users desire.

The possibility of TV viewing on-the-go as we commute from home to work and vice-versa, sharing a recorded movie with a friend or relative would be much appreciated and could add to the list of extra services an ISP would be able to offer.

We believe that such a future is possible and would like to discuss possible architectures and simulate the behavior of such a system. We also believe that the ISP's could decrease their operational costs, by taking advantage of the elastic properties of CC.

To be able to provide users with a quality service, ISP's have the need for computational resources, data storage and bandwidth that can follow the increasing number of subscribers. Streaming video or providing a VOD session to thousands and even millions of users require a lot of resources and it is here that CC comes in handy.

There is also the possibility of providing extra services since all the user data is concentrated in one ore more clouds that are under the management of the ISP. The new services could be for example Social TV where user's could interact, recommending specific channels and services to each other.

With the appearance of smart TV's that connect to the Internet we have a new way of accessing information and services in our living room. These devices have a lot of capabilities, allowing for network connection, video decoding, recording and human interaction.

We believe that the appearance of these devices have paved the way to de disappearance of the STB from our homes and the creation of virtual STB that are located in the cloud. This will lead to a decrease of the cost's ISP's have with purchasing and deploying these equipments, those funds will probably be channeled to the creation of new services and features.

1.3 Objectives

Our objective it to study and evaluate the possibilities that CC has to offer in terms of on demand growth and availability. We believe that these characteristics will enable CC to support the evolution of IPTV in the future. Our specific objectives for the elaboration of this work were the following:

- Survey IPTV and Cloud Computing to evaluate the possible synergies between a next generation IP based Television service and the Cloud;
- Modelling the IPTV Set-Top Box, namely the interaction with user and with the network. All the external interaction and internal processing and storage operations need to be modelled to have the complete design of all the used algorithms;
- Study the IPTV user behaviour based on existing literature to be able to synthesize user actions with an automatic log generator. As we have not access to STB logs owned by IPTV Service providers, we need to use a meta-analysis [15] of the related literature to generate our own logs;
- Design of cloudlets for each STB operation using the achieved models. These cloudlets will enable us to run our virtual STB on a cloud simulation environment;
- Based on the STB representation using cloudlets and the feed provided by the synthesized user activity, run a set of simulations on a Cloud Simulator;
- The simulation results, with a variable number of TV users, will give us an estimative of costs to support such Cloud based IPTV Service;

Smart TV's are becoming more and more popular, they have the ability to connect to the Internet and access social networking sites and VoIP services. We believe that they will also be able to connect to the cloud and make use of it's services. Since TV's do not have any storage or mobility capabilities we believe that CC will be able to help us with that.

Our vision (Figure 1.1) is to send the STB to the cloud and access it through a Smart TV, Smartphone or even a tablet. We want TV to be available anytime and anywhere, wether you are at home using your smart TV, commuting and using your smartphone to catch the latest episode of your favorite series, or even when visiting a friend or relative and show them on your tablet that movie you've enjoyed.

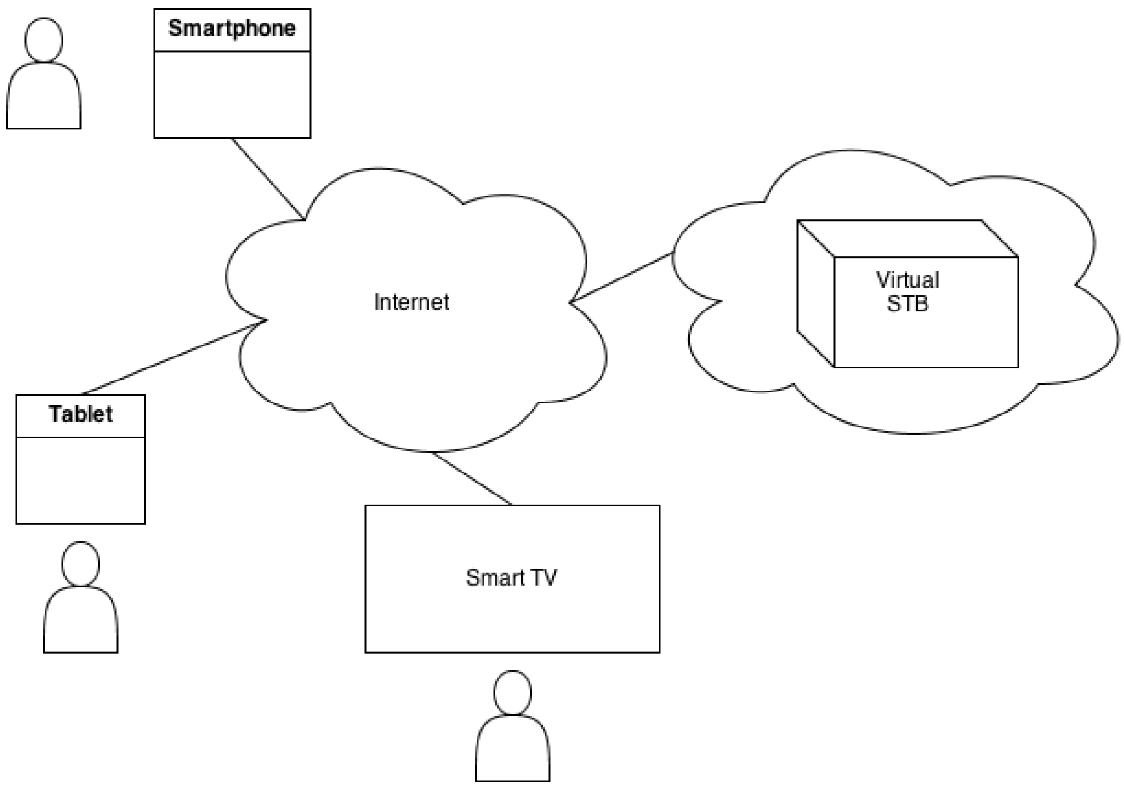

Figure 1.1: Our Vision of CC and IPTV

We want to move the STB to the cloud and take advantage of the possibilities CC has to offer in terms of resource escalation and storage management. By using a CC environment we can provide the service to a whole range of devices that can connect to the Internet.

For that to be possible we have to define and model the behavior of a STB and study it. We need to study the capabilities of CC and how does it behave during load and the mechanisms it offers to prevent resource starvation. We also have to map the interactions between the user, the STB and the network and analyze how they work together.

After having mapped and studied these activities we wanted to see how would it behave in a CC environment. For the creation of a real life test environment we would need to have access to log from ISP's with data regarding the behavior of user, but without that information available we had to improvise and create our own log generator. With the information about the activities a STB operates and the creation of logs to simulate a real life environment we did a simulation of how would a virtual STB behave in a cloud computing environment.

We have come to the conclusion that it would be possible to move the STB to the cloud but it would be expensive for a small amount of users. We believe that it would be less expensive if we took advantage of economies of scale for the procurement of a CC environment to deploy our solution.

1.4 Document Structure

Our objective is to uncover a small amount of the veil that covers the evolution of IPTV. We believe that the next logical evolution step for IPTV is towards The Cloud.

In Chapter 1 we make a small introduction to tour work and talk about the evolution of Internet and it's infrastructures and discuss the motivation and structure our work.

In Chapter 2 we start by discussing the way IPTV has evolved and the standards that are being developed. We discuss the architecture of IPTV and the way it is structured.

in Chapter 3 we study and model the behavior of users, how they interact with IPTV service and what are their utilization patterns.

In Chapter 4 we shift our focus to CC, we study it's origins and compare it to Grid Computing (GC). We present the different service and deployment models of CC and have a small discussion regarding the (in)-existence of standards.

In Chapter 5 we join the two previous subjects and propose a question regarding the possibility of CC being the future of IPTV. We also discuss the content delivery by using CC and talk about some of the simulation tools that can be used to model such a scenario. Finally we propose an architecture for the creation of IPTV in the Clouds.

In Chapter 6 we discuss the subject of STB, we model the possible user actions while interacting with the equipment. We create diagrams explaining the interactions and actions that take place while the users send commands to the STB

In Chapter 7 we start by discussing the utilization of the CC simulation tool that we have decided to use. We describe and show how a simulation can be created using CloudSim, we model the users activities to Cloudlets and create simulations that help us to gather some data regarding our initial proposition. The results of those simulations are then presented and discussed.

In Chapter 8, we discuss the results we have found and make some critical observations regarding the work we have developed. We propose some further work and present our concusions.

Chapter 2

Television over IP

In this chapter we wish to discuss IPTV. It can be seen as the natural evolution of TV, in the Internet era. We should point that IPTV is not the same as Internet TV, it is delivered over an ISP's infrastructure that has been carefully thought(Figure 2.1), so it can provide the best QoS to it's clients. Internet TV is just a service that can be used through the Internet so that users can access media content streamed over the net.

In order for an IPTV service to be provided, the ISP's network infrastructure has to be carefully though off, components like Head-end encoders and Digital Rights Management (DRM) servers, among others, should be available. IPTV is quite bandwidth demanding, for each TV channel it needs approximately 2 Megabit per second (Mbps). Now imagine that a ISP offers 100 channels, this means that the backbone should have at least 200Mbps The network should also be reliable because if you miss any frames during a TV show you might loose some important part of it, the same can be said for audio.

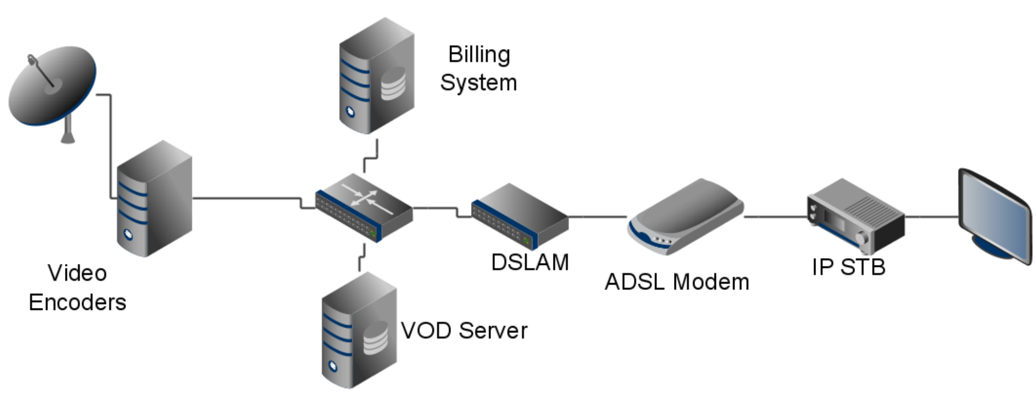

Figure 2.1: Basic IPTV Architecture.(adapted from [1])

When we look at Figure 2.1 we can have an idea about how an IPTV service is composed of. On one side we have the Video Encoders, that obtain the TV shows and movies and encodes them so they can be sent to the clients through the network or stored on the Video On Demand (VOD) Server for future renting.

The streams that represent the TV shows or movies a user might rent are all sent from the ISP's network to the Digital Subscriber Line Access Multiplexer (DSLAM) near the user's house and from there are distributed to the subscribers of each channel through their network access interface which is the Asynchronous Digital Subscriber Line (ADSL) Modem, but not before having authenticated on the Billing System that monitors the access of user to the service.

The final step is made from within the user's hour, the stream is collected and organized by the STB and presented through the TV set. All this transmission is made with great care and with QoS measures that prevent the failure or improper functioning of the service.

2.1 Next Generation Networks

When working with technologies that evolve so rapidly as networking, one needs to be aware of the current standards, for that to occur, there is a need to identify the main players when it comes to IPTV and networking infrastructures.

The compass points towards International Telecommunications Union (ITU) and European Telecommunications Standards Institute (ETSI) - Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), as the main players regarding IPTV and it's complementary technologies. IPTV is supported by IP Multimedia Subsystem (IMS) [2] which is supported by NGN[14]. Both of the organizations have created Focus Groups, in order to discuss and create worldwide standards.

In networking, NGN is the buzzword that is heard the most nowadays, this is a term used by the networking industry, to summarize all the technologies and mechanisms that will be implemented within the next years. The concept of NGN is to be able to transport every kind of information, whether it's video, audio, data or even voice through a single network. NGNs are the evolution of both networking infrastructures and protocols, they provide the convergence of networking services and availability through it's several interfaces.

One of the most challenging aspects of NGNs, is to develop means of evolving the current Public Switched Telephone Network (PSTN) lines that have been deployed by Telecommunications Operator (Telco). Not all the networks have the same sort of infrastructure and services, so the ITU [14] and ETSI-TISPAN [2] created a Focus Group responsible for all aspects regarding NGN.

The ITU's NGN Focus Group (FG) and ETSI-TISPAN were responsible for the creation of the NGN Standards, so that Telcos would be able to keep up with the momentum regarding

NGN.

The future of NGN will most certainly pass through the implementation of FTTX infrastructures, that will allow Telcos to provide greater bandwidth and increased QoS to their costumers. Portuguese Government is promoting the usage of optical fiber throughout the country, and is boosting the creation of infrastructures so that NGN can reach most of the population.

The usage of optical fiber as a mean of access by Telcos, resides on the fact that it can provide greater bandwidth and also carry the signal farther. It is more reliable, because it doesn't suffer from any kind of interference. It is more secure, because it can not be tampered with, and finally it has a longer life span compared to copper wires. Despite having all the previous mentioned qualities, optical fiber has the disadvantage of being more fragile and prone to rupture if not handled properly.

2.1.1 IP Multimedia Subsystem

Since NGN's access architectures have been referred, we should also mention the main functional architecture used for delivering multimedia services within the NGN context. The afore mentioned architecture is called IMS, developed by ETSI-TISPAN and the 3rd Generation Partnership Project (3GPP) [2, 3] . This architecture, is one of the steps towards the All IP network: It allows for the integration of several services within a network infrastructure. One of it's most significant aspects is that it does not care about the access technology used to connect to a specific service. It seamlessly provides that service taking care of the QoS aspects, and adaptation methods for each single connection type.

IMS Architecture

The TISPAN IMS architecture, is defined by three layers [2]: the application layer, the control layer and the transport layer, as can be seen in Figure 2.2. The application layer, consists of the application servers that host the IMS services and the home subscriber server. The control layer is composed of several systems, among them the IMS Core.

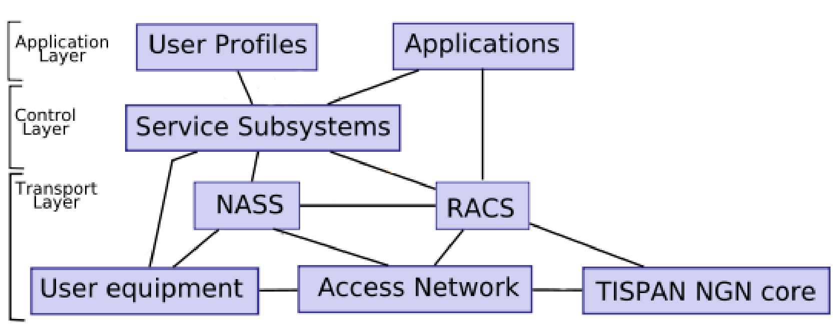

Figure 2.2: TISPAN IMS Architecture.(adapted from [2])

The transport layer consists of the User Equipment, the Access Network, the Resource Admission Control Subsystem (RACS) and the Network Attachment Subsystem (NASS).

IMS is mostly used for flow and session control [2].

In order to work, IMS needs two other non standard subsystems: RACS and NASS. These two systems are responsible for dynamic configuration, authentication and admission control [2, 3] . IMS uses several standard protocols for session establishment, authentication and policy support. Protocols like Session Initiation Protocol (SIP) [16], Diameter[17] and Common Open Policy Service (COPS) [18], together with the IMS core functions, transform this solution into an all-in-one package, for network management and implementation [2] . One of the characteristics of IMS is that, it provides seamless service integration on network infrastructures. Nowadays, one of the most requested services is IPTV, and IMS has also a solution for it's integration.

IMS allows for the possibility of user mobility and service access by multiple equipments through different network types.

Translating this to the diagram we see in Figure 2.2, a User who is viewing an IPTV stream (User Profiles, Applications) from his smart TV (User Equipment) wishes to leave his home and continue viewing the same show on the bus. He stops the show on the smart TV and uses his smartphone(User Equipment) to connect to this IPTV provider with the same user and password.

Here is where the Service Subsystem kicks in, by becoming the intermediary between the service the user is accessing through one of his equipments and a specific kind of Access Network.

The Service Subsystem uses the NASS to provide the User Equipment with the configurations to use the new type of Access Network. Information is also sent to the RACS so it can adapt the IPTV service to the new kind of Access Network.

All of these adaptations and service preparations are managed by the Service Subsystem which sends the new adapted stream from the RACS to the users smartphone so he can continue viewing the same TV show(User Profiles, Applications).

This kind of services that allow for the adaptation of the service to any kind of media it is being access through will allow for the improvement and appearance of new services and ways to interact with networks.

One of the characteristics of IMS is that, it provides seamless service integration on network infrastructures. Nowadays, one of the most requested services is IPTV, and IMS has also a solution for it's integration.

2.2 IPTV Architecture

In order to understand how IPTV works we have to review it's architecture and the way all the components work together to provide the service. In the early stages of IPTV the available solutions did not take in consideration the NGN subsystems and were focused on vendor specific products that did not allow for the integration of services.

With the advent of IMS as a proper standard, ITU and ETSI -TISPAN included it as a part of the NGN standard. Providers were deploying network infrastructures without looking into the standards and something had to be done if Providers wanted to be able to keep up with the evolution of IPTV services.

With this idea in mind Mickoczy et al . [3] decided to define an evolution of IPTV architecture towards the NGN. They proposed to reach this goal with four major steps(Figure 2.3): the first is a non NGN IPTV, the second is a NGN non IMS IPTV , the third a NGN IMS IPTV and the fourth a NGN converged IPTV.

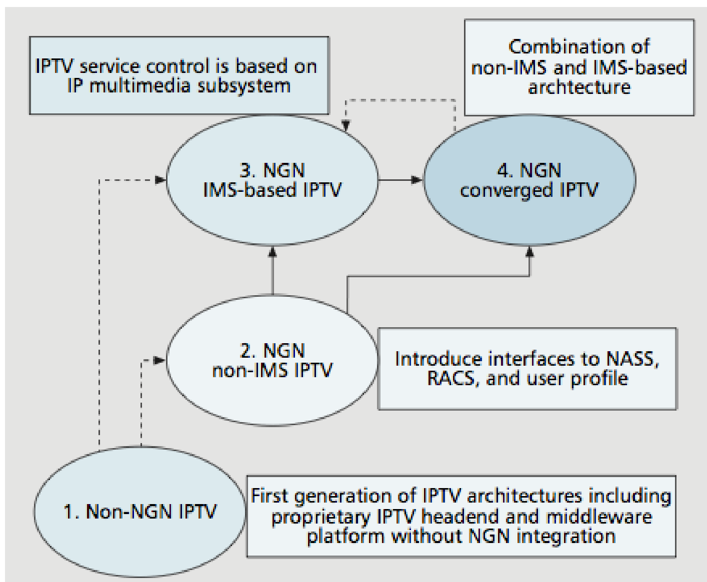

Figure 2.3: IPTV Architecture steps.(extracted from [3])

Each one of these steps explains the path that should be made in order to create a converging our IMS based NGN IPTV solution.

The first step is made up of all first generation IPTV implementations that where made without regards to standards and norms. These were vendor specific implementations of IPTV without any regards for standards and interaction with other services.

The second step introduced components like NASS and RACS for the control of users and profiles and access to services. These implementations took a little more care in thinking about standards and interactions by introducing components that allow for a better management of resources and interaction with other services.

The third step is a implementation of IPTV based on IMS. This is the desired evolution for all IPTV systems, allowing for the adaptation of the service to the kind of network it is being used and also allowing for the mobility of users between networks and equipments.

The fourth step is the integration of IMS and nonIMS IPTV architectures, since not all providers will change their systems to behave in a standardized manner. This provides a way standardized IPTV services to interact with non standard IPTV services.

In the future not all IPTV implementations within NGN will be IMS based, that's why step three and four are at the same level, meaning that not all Providers have the money or the technical expertise to create a full standard architecture. What they can do is create an architecture that can coexist with others that apply the standards.

Another view of the IPTV architecture is given by the ITU-T FG, and it describes that there can be three approaches [4] to IPTV architectures: non-NGN , NGN non-IMS and NGN IMS.

The afore mentioned architectures are quite similar to the ones previously mentioned. By the ITU perspective all of the IPTV implementations will undoubtedly be ported to an NGN IMS based IPTV architecture.

In their definition of an IPTV architecture the ITU defines four major functional domains as can be seen on Figure 2.4.

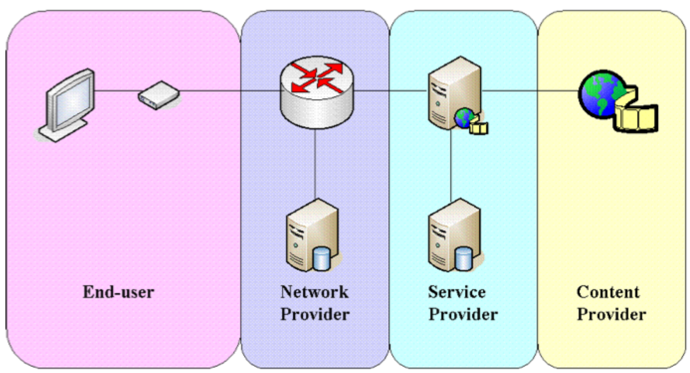

Figure 2.4: ITU-T FG IPTV Architecture.(extracted from [4])

The Content Provider owns or sells the media content, the Service Provider is a Telco that offers services to the end user, the Network Provider is the one that takes care of all the network infrastructure(he can also be the Service Provider), the End-user is the entity that accesses the network to obtain services.

We have also found a article by Silva et al . [5] that describes the basic IPTV architecture elements. We believe that it is a very clear explanation of what every component should be responsible for. IPTV depends on four basic components; the Video Head-End, ISP Core Network, ISP Access Network and Costumer premises [5]. Each one of them will be described on the following subsections.

2.2.1 Video Head-End

The Video Head-End is the place where it all starts. In this part of the system video signals and all the media that is to be distributed over the network are acquired. The digital and satellite TV signals are collected through satellite dishes, digital antennas, connections to media providers or direct connections to the TV Channel's network. It is also the place where the on-demand content is also stored/gathered so it can be sent out to the costumers. Another main feature of this component is the codification of all the media in to a digital format such as MPEG-2/4. This is done so it is possible to encapsulate media into IP packets that are to be sent through the ISP's network. The main codec used in the early IPTV solutions was MPEG-2[5], this solution allowed for the streaming of video over networks with very low bandwidth because only needed a 3Mbps link to successfully stream video.

With the appearance of h.264/MPEG-4 AVC [5] the streaming of video over the network became easier, due to the fact that it allows the coding of video with high quality at lower rates. It can be used in a lot of equipments and applications and also, with the inclusion of Network Abstraction Layer (NAL), it can adapt its coding according to the network type it will be streamed through. As we can see in Figure 2.5 MPEG-4 AVC allows for the availability of more channels per connection.

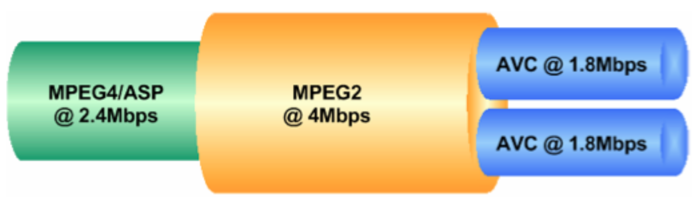

Figure 2.5: MPEG-4 AVC vs MPEG-2.(extracted from from [5])

By reducing the need for bandwidth, MPEG-4 AVC facilitates the existence of extra channels for the same original bandwidth. Also by reducing the bit rate of the encoding process it also increases the distance a specific stream can travel through the medium.

2.2.2 ISP Core Network

The ISP Core Network is the place where all the magic happens. Here all the data that was previously acquired and encoded into MPEG-2/4 is encapsulated into IP packets and passed on to the ISP's access network.

In order for the service to be provided with the best quality possible and to the highest possible number of costumers, the data is sent using multicast [19] IP communications. By using multicast communications the ISP allows the end user to join (tune) a specific multicast group so that he can choose which channel/groups of channels we wishes to view.

Multicast also allows for scalability a reduced latency and load of the ISP's servers while tuning into a new channel.

The usage of the Internet Group Multicast Protocol (IGMP) [20] management is used because it allow for the ISP to manage to whom he wishes to send the content within the network. If normal point-to-point connections where used in IPTV services we would have a waste of bandwidth because every client would have to make a direct connection to the content he wanted to access. this would create a flood of requests through the ISP core network and could probably throw down the network.

By using different groups the ISP can filter content by simply allowing or disallowing some users access to it. Every time a user wishes to tune a specific channel, his STB sends an IGMP join message and the equipment is given access to the feed that is being transmitted through that specific group. When a user wants to change channel his STB sends an IGMP leave message followed by another join message so that the process of gathering the stream from other group is able to start. Also by using IGMP,

2.2.3 ISP Access Network

This part of the system is commonly named as the last mile, because it's the last component between the user's home network and ISP's network. The distribution of the services within the ISP's Access Network can be accomplished by using a variety of technologies such as XDSL, FTTX and mobile networks.

This network delivers the encapsulated content to the edge of the network where special equipments such as DSLAM or Optical Line Terminal (OLT) are ready to provide it to the end user's equipment.

IPTV requirements are greater that the ones needed by traditional services, costumers expect the service to reach them without any failures or hickups.

IPTV requires a minimum bandwidth of approximately 3Mbps for Standard Definition (SD) video and 8Mbps for High Definition (HD) streams [6]. It is stated in [6] that the "magic number" for a successful IPTV service is in the range of 20-30Mbps, allowing for 1 HD channel and up to 3 SD channels.

The usage of Fiber To The Node (FTTN) or Fiber To The Building (FTTB) technologies together with XDSL allow these kind of speeds to be available to the end user.

XDSL is also evolving and now can enable higher bit rates and bigger service coverage.

ISP's are continuing to evolve access network infrastructures to be able to present high quality services to their clients. These infrastructures include deployment of: FTTN where a DSLAM is placed on a street cabinet and the last mile connection is made with XDSL; FTTB where a small DSLAM is placed on the basement or entrance of a building and the rest of the connection is made using XDSL;Fiber To The Home (FTTH) where the fiber is sent directly

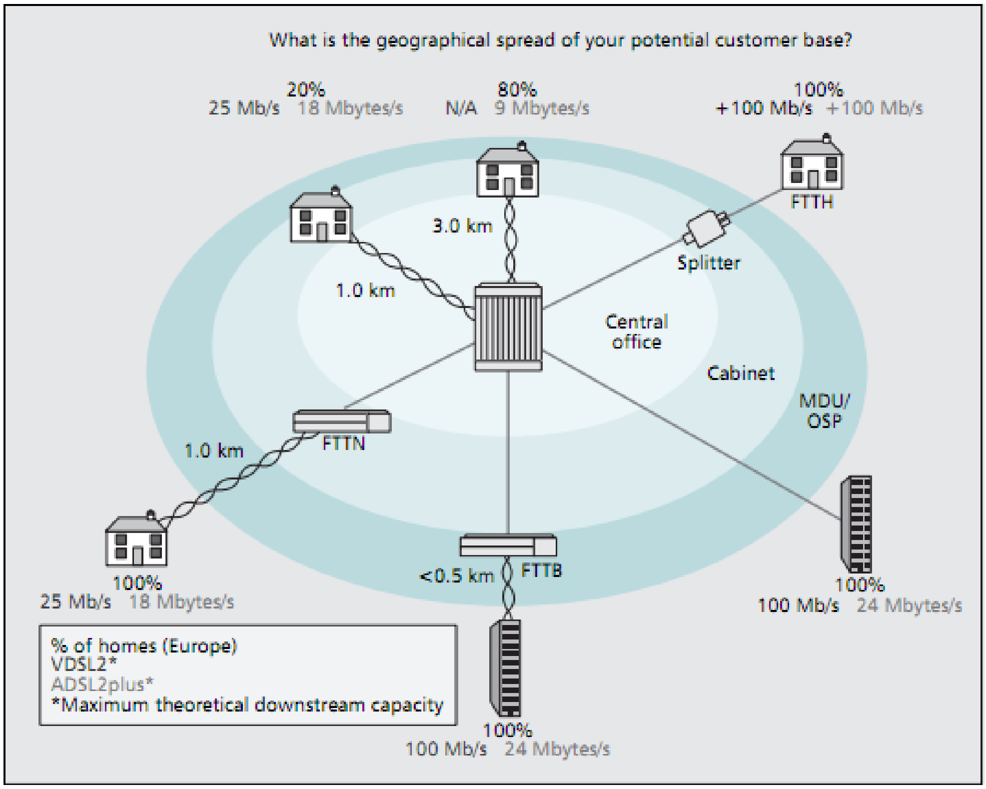

Figure 2.6: Access Network .(extracted from from [6])

to the costumer's house.

With these different types of infrastructure deployment ISP's can offer different speeds and ranges to their services as can be seen in Figure 2.6 .

FTTN allows for speeds of 18Mbps at distances of about 1Km,FTTB allows for speeds of 24Mbps at distances of about 0.5Km and FTTH allows for speeds of 100Mbps.

As we can see in Figure 2.6 the FTTN and FTTB speeds are lower that the ones of FTTH, this is due to the fact that the optical fiber is sent from the ISP's central office into a concentrator and then the signal is shared between the users of the service. In FTTH implementations the optical fiber is sent directly into the client's house, hence the higher speeds.

Despite not being the most effective in terms of bandwidth FTTN is still the deployment mode of choice for most ISP's because it is cheaper and can still do the job.

2.2.4 Costumer Premises

The Costumer Premises is where the client's network equipment and STB is located. The network equipment connects to the ISP's Access Network and serves as the middle man between costumer's home network and it's STB.

In this part of the system the IP packets are received from the specific multicast group the user has joined/tuned in the STB. These packets are then and decoded so that they can be viewed by the end user.

The distribution of the content through the house can then be made using a variety of technologies such as coaxial cable, wireless networks or even ethernet.

2.3 IPTV Service Quality

QoS and Quality of Experience (QoE) are two concepts that must considered when it comes to IPTV.

While QoS refers to the Quality of the service provided by the underlying network protocols and equipments in IPTV infrastructures, QoE is the perceived quality of the end users.

Both of them can be measured and classified but QoE is much harder to quantify once it is mostly perceived by the end user and not by the equipment.

An ISP network can be projected in such a way that all the requirements regarding packet loss, jitter, bandwidth and load are met and still provide a bad QoS regarding IPTV. The packets can all arrive at the desired moment and all the bandwidth requirements can be met and still the user can complaint about QoE.

In order to provide a good QoS and QoE to the end user, the network must provide transport-layer and service layer performance [21].

Some state that in order to have QoS and QoE one must employ some kind of signaling [22] in the network for resource allocation and monitoring, others say that there should be a more vertical approach [21] to the issue of QoS and QoE by implementing NGN-Based IPTV.

There is also an approach to the problem of QoS and QoE by combining both in a layered model[23]. The main layers are Service, Application and Transport/Network Table 2.1.

QoE Domain	User
	Service
QoS Domain	Application
	Transport/Network

Table 2.1: Qos/QoE Layered Model

The Service layer represents the part that is exposed to the user and the part where the QoE is measured, by the opinion of the user while enjoying the service. The Application layer, as part of the QoS domain is related to parameters of video application such as resolution, frame rate, color, codecs and the way they are implemented so that a specific QoE is achieved.

The Transport/Network layer regards parameters as jitter, delay and packet loss. By using this approach we can manage two ”problems” as one but still specify the needs of each one of them.

By using this method of QoS/acQoE management we can do a top down approach, first we start by identifying the QoE requirements, then the QoS requirements and finally the technology architecture.

If one is planning to have QoS and QoE, one must monitor the service and be sure that all the implementations that have been deployed are woking properly. The monitoring of video quality can be done in several ways, it can be done by using subjective methods and objective methods, but always in search of impairments such as blurring, color errors, jerkiness, object persistence, object retention and lip sync among others.

Subjective methods require that a group of people watch a video and give it some scoring, this method is not the most practical one to be used in live systems. Objective methods use a model of human quality perception implemented in some sort of automated test that monitor an IPTV deployment.

We believe that there is also another aspect that can improve both the QoS and QoE, and that it the usage of proper codecs. If one uses an H.264/MPEG-4 [24] codec instead of MPEG-2 as stated in [5] we can gain both a better compression rate and service better service coverage. In this field there is also an evolution, with the availability of H264/MPEG-4 SVC extension[25], the encoding and decoding of video allows for adaptation to the medium in which it is transported, wether it is XDSL,FTTX, or even Wifi.

H264/MPEG-4 SVC extension improves QoS and QoE by optimizing content for distribution, it allows for de usage of older and newer equipment to decode stream by simply discarding the layer that has not been implemented on their codecs. By doing so media content can be streamed using H264/MPEG-4 SVC and viewed on equipments that implement older codecs that are compatible with SVC.

2.4 Summary

In this chapter we have discussed IPTV, it's architecture and the way all it's components interact, we have seen that the major struggles for this kind of service is the definition of standards that can be used to give the final users a more satisfying QoS and QoE. But some evolutions are on the way to help us reach semi-standard implementations of IPTV services and interaction between its components.

Implementations like IMS will allow for the interaction of for services and for the adaptation of the services to different kinds of networks and equipments.

The evolution of new ways of encoding and decoding IPTV streams will also allow for an improvement in both quality and availability of the service.

Finally the transport media evolution will also bring some possibilities for the growth of bandwidth allowing for quality and speed at the same time.

Chapter 3

IPTV User Behavior

Television has been the main media since the 1950's, it has come a long way since the black and white broadcast channels. It has been subject to many technological advances, with IPTV being the latest one.

Now we have the opportunity to offer services with a better quality to the users. If one want's to improve the QoS and QoE it is necessary to get to know the user's behavior, and that is what we will be doing in this chapter.

We need to know how the user interacts with the system so we can provide him with a better experience, while reducing the usage of resources.

Each user, being a unique individual has its own behavior, it likes specific programs, content and dislikes others, he views television alone or in a group, he lives in a specific part of a country and has it's own viewing habits. It is quite difficult, maybe even impossible, to create a profile of each user, that's why it is easier to create global user profiles.

If we can create several profiles in which our users can fit in we are capable of creating specific content and provide it to them. With these profiles in our possession we are able to view user habits and provision our resources accordingly. In order to map the user's behavior their usage of the system has to be analyzed and interpreted [8, 7] so that patterns can be found and studied. This can be done by means of meta-analysis [15], through the combination and analysis of results from other studies.

We can take some advantage from the utilization of the IP protocol, being a bidirectional mean of communication it can be used as a tool to infer user activities. With this information at hand we can take IPTV to the next level of QoE and QoS, while also taking advantage of this information to design and improve our infra-structure and media access.

3.1 Active Users

The usage of IPTV systems is not the same throughout the day and also the week, the system isn't always loaded with requests and streaming content to every household. In [8] and [7] despite having very different dataset sizes, 350 households versus 250.000, they have come up with very similar patterns of IPTV usage, giving us data regarding the number of users in a given period of time.

In [8] we have a universe of 350 households of a Swedish municipal network that provides IPTV services to their users. They have found that the average number of people using the IPTV service had a distinctive pattern. The usage of the network was low during the morning and started increasing during the afternoon, peaking at about 8PM and then decreasing throughout the night both during weekdays and weekends (Figure 3.1).

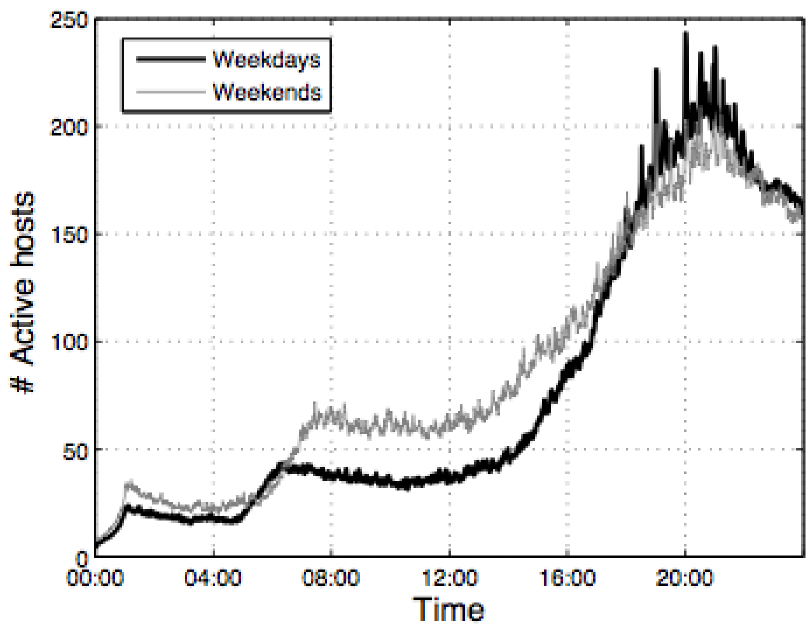

Figure 3.1: Active Users (extracted from [7])

In [7] we have data from a commercial IPTV supplier with approximately 250.000 users, despite having a dataset several orders of magnitude higher than the previous one, the same distinctive viewing pattern arises(Figure 3.2), uses a much more active during the afternoon and around dinner time and then start plunging throughout the night.

We can identify this as being the pattern the most of us follow during our working week. We wake up early and turn the TV on to catch the news, then we go to work and by lunch time we take a look a the news once again. In the middle of the afternoon the kids get home and start watching their favorite cartoons, when we get home the kids start doing their home work and the TV is turned of. When we finish dinner the TV is turned on once again and is kept on until we go to bed.

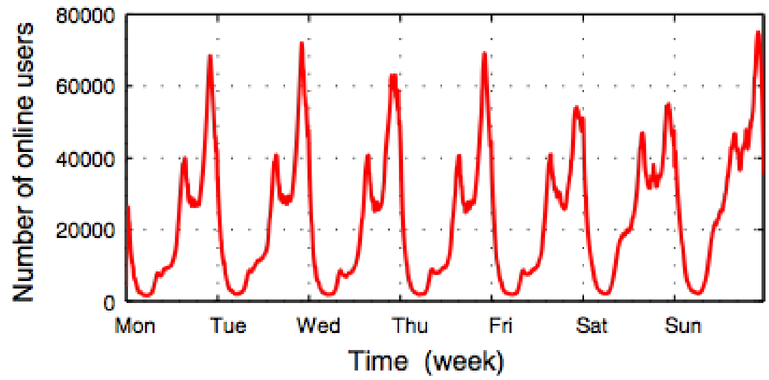

Figure 3.2: Active Users (extracted from [8])

3.2 Session Length

We have reviewed the data regarding the number of users we can also review the data regarding session length, in both [8] and [7] they have found that user sessions that last longer are in lower quantity than the sessions that last for a small amount of time. This is related with the zapping time. Users tend to have lots of short sessions and only a few long ones.

In [7] they have realized that the session time increases from 1 to 4 seconds and then decreases very rapidly. This is due to the fact that 4 seconds is enough to perceive the TV channel and make the decision to view it or to change the channel. As we can see from Figure 3.3have the results from session length and from them we can infer that the huge amount of sessions that are between 1 and 4 seconds are due to zapping, the number of sessions that take more than 4 seconds are related to the interest of the users to continue viewing the same channel.

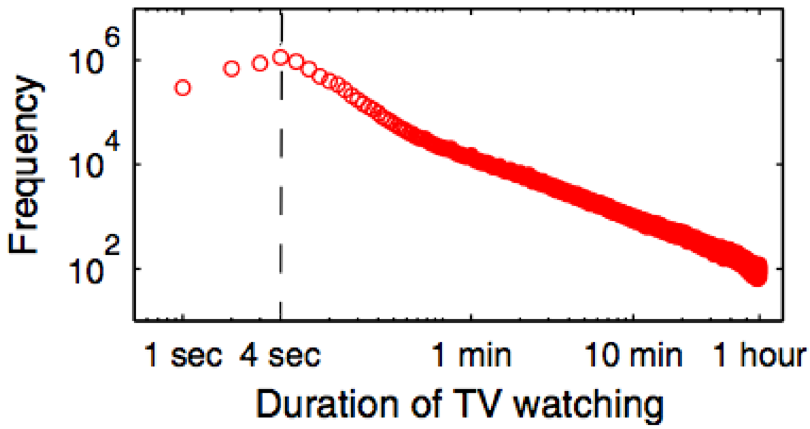

Figure 3.3: IPTV Session (extracted from [7])

Despite having a substantially lower dataset, in [8], we can also view in Figure 2.12 the same behavior regarding session length, there are a lot of sessions that last less than one minute and the sessions with a length of more then one hour are a very small number. Like we have said previously, here the users also do fast zapping in order to reach the desired channel.

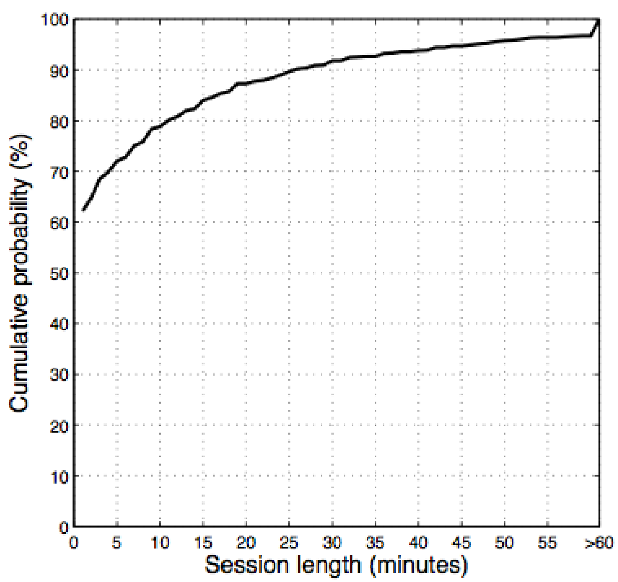

Figure 3.4: ITPV Session (extracted from [8])

3.3 Channel Zapping

With all this data in hand we can conclude that most of the sessions that are started by users are transitional, probably due to the fact that in order to reach the desired channel one passes through a bunch of other ones that are in between.

This kind of behavior leads to the creation of a great number of meaningless usage of resources. The act of switching between channels triggers IGMP leave and join messages and forces the network equipment and the whole infrastructure to start a new session and prepare for a different stream from behalf of a new IGMP group.

In [26] they propose a solution for lowering the usage of resources and increasing the speed while changing channels. Their solution is based on the utilization of H.264/SVC which is an extension of the H.254/AVC, the later is composed of two parts, the base layer and one or more enhanced layers.

Normally when a user wishes to change a channel he uses the up and down buttons until reaching the desired channel. What if we could pre-join the channels the user will go through in order to save network resources? This is where the H.264/SVC come in handy. We can pre-join the base layer of a number of channels so when the user switches channels the stream is already at the equipment. When the user selects a specific channel to view, the base layer is the only one he has available and that is enough o start viewing the desired program, the STB just has to join the enhanced layer to provide for a better QoS. This is done so because the base layer consumes 10 times less bandwidth than the overall H.264/SVC stream.

In [27] there is also a proposition to minimize the zapping time by pre-joining channels, but in this case they only pre-join channels during the surfing period unlike in [26]. The proposal from [27] is to pre-joing only a small number o channels during a specific period of time. They decided to compare the pre-joining of 2,4,6,8, and 10 channels, for the period of 10,30,60 and 120 seconds with an ideal predictor and view the percentage of switchings with no delay. The ideal predictor provides almost no delay as it would be expected, but the pre-joining of only 2 neighbor channels for only one minute the delay is almost zero for 45% of the switching events as it can be seen on Figure 3.5

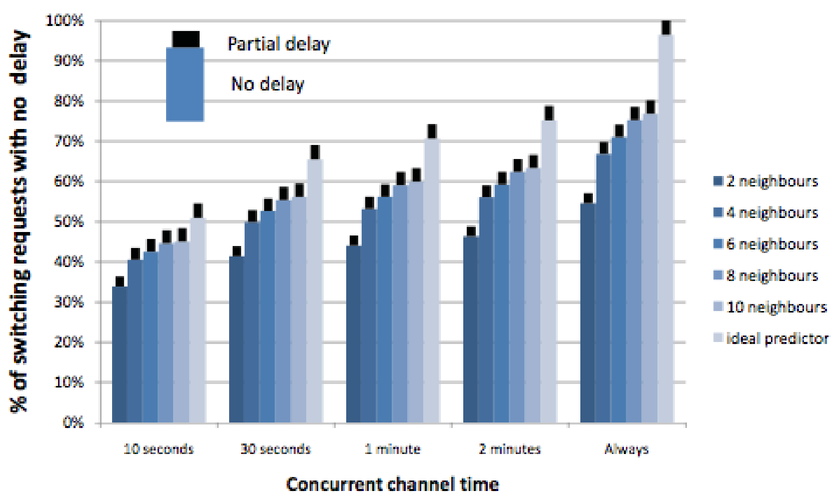

Figure 3.5: User Zapping (extracted from [8])

3.4 Modeling User Behavior

It is quite difficult to model the user behavior without any consistent data, one can estimate how the user behaves by extrapolating our own behavior and activity as users of IPTV services. While sometimes this can help us reach a significant knowledge regarding other areas, when it comes to human machine interactivity this is not always the rule to follow.

The best way to learn and know how a user behaves and interacts with an IPTV system is by accessing the data they produce while using the equipment within their homes. Most of the IPTV STB's have a log mechanism that sends data back to the ISP for analysis and this information can be used to infer how a user interacts with the system and what are the patterns of activity he generates. This information can be later analyzed and used to improve QoE and QoS.

3.4.1 STB logs to the rescue

In [9] and [10] the authors have used such information from real ISP's and used it to model and plot the user behavior, giving us a glimpse of how the users interact with an IPTV system. They have gathered logs from the STB's to try and figure out what are the most common actions or options used by user while subscribing to an IPTV service. Despite having the same objective in mind the authors have decided to engage the problem in different ways. Silva et al. decided to gather statistical information and plot it. Qiu et al. also plotted the data but created mathematical models and a simulator of workloads to help study the user behavior.

3.4.2 User Activities

Contrarily to what one might think, the top activity that is logged by the STB's isn't channel change, according to [9] it is the 6th most occurring event, the activity that is logged the most is the SMSR. After having discovered which are the top activities that are logged by the STB, Silva et al. have decided to choose a set which reflected the user actions and not those of the STB itself. In Figure 3.6 we can se the top activities that where registered in the logs that where studied, we can also see if the where originated by the user or the STB.

Event ID	Event Description	Originator	Network impact	%	Notes
105	Browse Panel	User	Unicast starts (channel browse, PIP)	14,00%	User browses through PIP version of Live channels
100	Channel Tune.LIVE	User	Unicast + Multicast starts	5,20%	User tunes to a Live channel. ICC occurs: unicast burst plus multicast join.
100	Channel Tune.VOD	User	Unicast starts	0,40%	User tunes to VOD channel
104	Trick State.VOD	User	Unicast pauses, resumes, etc	0,30%	User acts on VOD stream (pause, play, rewind, etc)
17785	Stream Management Stream Request.VOD	STB	Unicast is ongoing	0,10%	STB requests while user is tuned on a VOD channel
17781	Stream Management Detune.VOD	User	Unicast stops	<0,01%	User tunes away from stream. Has small statistical relevance
17785	Stream Management Stream Request.LIVE	STB	Multicast is ongoing	42,50%	STB requests while user is tuned on a Live channel
104	Trick State.NEWLIVE	User	Multicast is ongoing	2,00%	User acts on Live stream (pause, play, rewind, etc) while STB continues connected to the multicast group
115	DVR Start Recording.ByUser	User	Multicast starts/continues	<0,01%	STB joins multicast if not already connected (Live). ICC does not occur.
115	DVR Start Recording.BySystem	STB	Multicast starts/continues	0,60%	STB joins multicast if not already connected (Live). ICC does not occur.
17781	Stream Management Detune.LIVE	User	Multicast stops	0,03%	User tunes away from stream. Has small statistical relevance

Figure 3.6: STB Activity (extracted from [9])

The events that where considered for the study where Chanel Tune, Browse Panel, Trick State, SMSR, Stream Management Detune and DVR Start Recording. So what do these events mean and how significant are they to define how a user behaves?

3.4.3 Chanel Tune

Channel Tune provides us with the information of the tune/join of a specific channel, it can be a live tv, a VOD or DVR channel. The tune of a live tv or a VOD channel implies that the user has to receive a new stream from the network, while the DVR stream video locally. The Channel tune event is lower during dawn, it increases during the morning until it peaks at about 13:00h and then slightly decreases during the afternoon and peaks again at 21:00h (Figure 3.7).

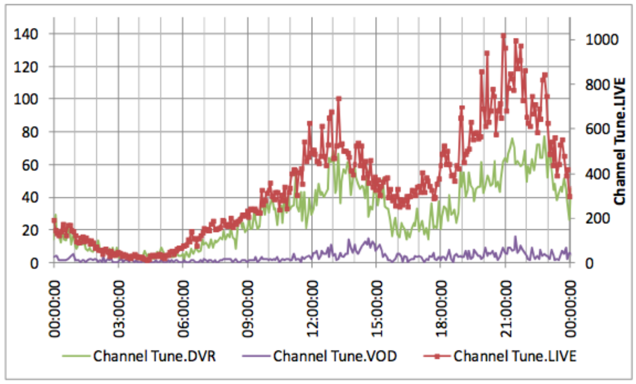

Figure 3.7: Channel Tune Events (extracted from [9])

This is quite logical because people tend to view TV when they have spare time and that is around lunch and dinner time. With the data gathered in [9] we can see that 50% of the events are shorter than 8 minutes and 90% are shorter that 1,5 hours(Figure 3.8)

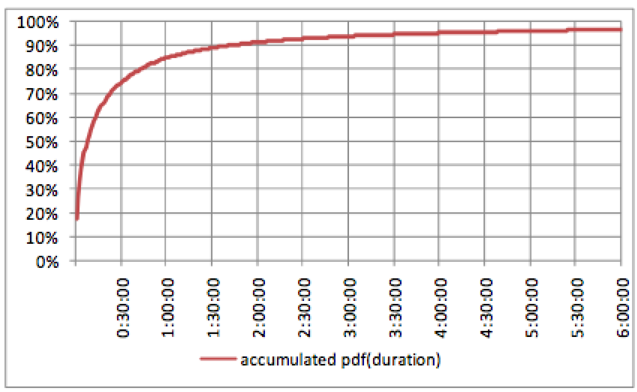

Figure 3.8: Channel Tune Duration (extracted from [9])

This leads us to the conclusion that for most part of the day the system has a very low channel change and it peaks to about double at lunch and triple at dinner time. This doesn't give us enough information to deduce that the System is in high load because there can be a lot of "tune-in's" and very low viewing time. For us to reach a conclusion about this matter we must view the other results.

3.5 Browser Panel

The Browser Panel event occurs when the user navigates through the Picture In Picture (PIP) version of the channels that the ISP provides through its STB Graphical User Interface (GUI). The PIP visualization of a channel can help the user decide which channel he will tune into next. Like the previous event this one also follows the pattern of peak activity during lunch and dinner time(Figure 3.9).

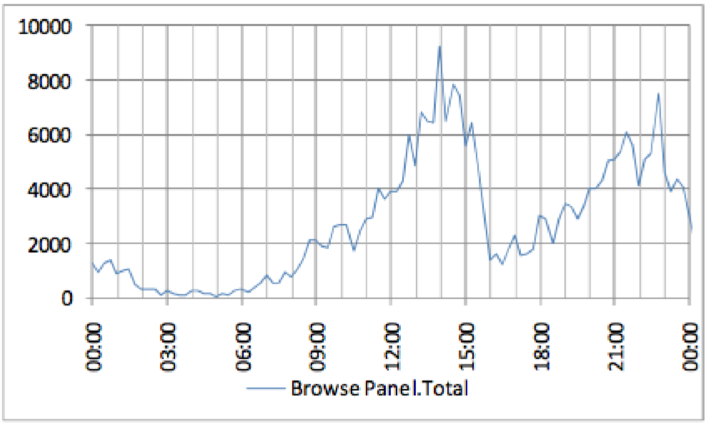

Figure 3.9: PIP viewing pattern (extracted from [9])

This data is helpful because it can help us provision for Instant Channel Change (ICC), by analyzing Channel Tune and Browser Panel together we can infer the ICC rate and demand across the day.

3.6 Trick State

Trick State events regard the user actions like pause, play and rewind wether it's made during DVR,VOD or Live TV viewing. When Trick State regards to Live TV and DVR they don't imply any special synchronization because the content is already within the costumer's premises, but when it comes to VOD that's not the case.

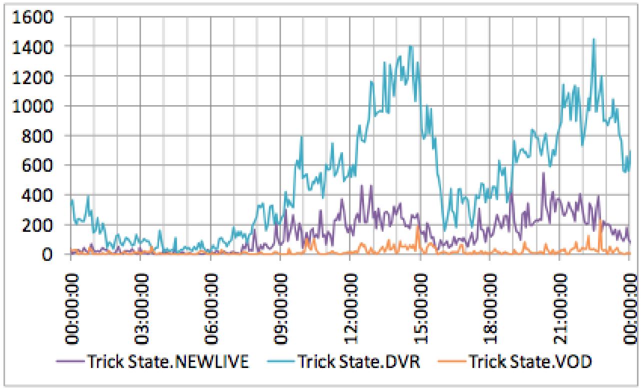

Figure 3.10: Trick State pattern (extracted from [9])

Trick State applied to VOD increases the load on server because it needs to sync the streaming of content according to the users actions. As it would be expected the majority of trick states regards DVR(Figure 3.10), because people use the STB's recording capabilities to save shows that they can not view during the day or at the same time.

3.6.1 SMSR

Stream Management Stream Request is the event that occurs the most in the logs(Figure 17). Although it is not related to any user action in specific it directly related to the state of the STB. When the user is viewing Live TV, catching up some past shows on it's DVR or viewing some movie through the VOD service, SMSRs takes place. This happens between the STB and the Server(VOD/Live TV) or between the STB's that are spread across the user premisses(DVR). By analyzing Figure 3.11 we can see that the SMSR follows the same pattern as the Channel Tune and the Browser Panel activities, confirming once more the user's IPTV viewing/usage follow and activity pattern.

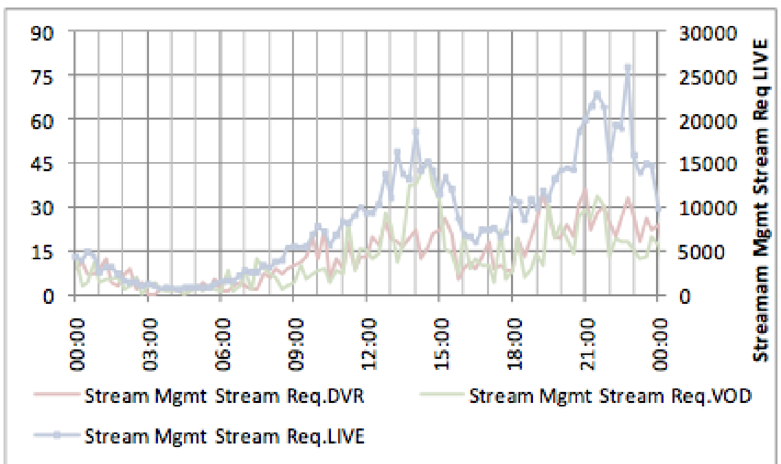

Figure 3.11: Stream Management Stream Request pattern (extracted from [9])

By analyzing the work made by Silva et al. we have concluded that the user's activity follow the patter of moderate usage during dawn, this usage increases during the morning and peaks at 13:00h then it diminishes slightly during the afternoon and at around dinner time it starts increasing and peaks at 21:00h. In [10] Qiu et al. studied the on and off sessions of STB, Channel Switching, channel popularity and modeled Session length, channel popularity distribution and dynamics.

3.6.2 On and Off Sessions

An On Session is defined by the time a STB is switched on until it is switched off, similarly an Off Session is the period between the last time a STB was switched of until it is switched on back again.

Qiu et al. studied the length of both On and Off Sessions and discovered that around 5%(Figure 3.12)of the On and Off Sessions where higher than 1 day long. and that the Off Sessions had a heavier tail.

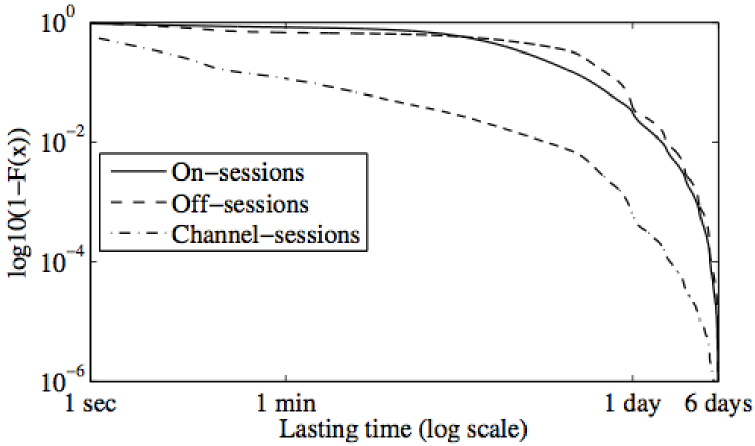

Figure 3.12: On and Off Session Length (extracted from [10])

This is logical because a user normally leaves the TV turned off for longer periods than turned on. If nobody's watching why should it be on? In this study they have also came across the same usage pattern the we have previously seen in [9].

In figure 3.13a/b we can see that during dawn there isn't a lot of activity, but then in the morning users start turnig on their STB's mostly for news updates and checking the weather, then activity start reducing and stabilizing until lunch time when it peaks just little bit, it keeps steady and then peaks again at dinner time, after that is starts decreasing throughout the night.

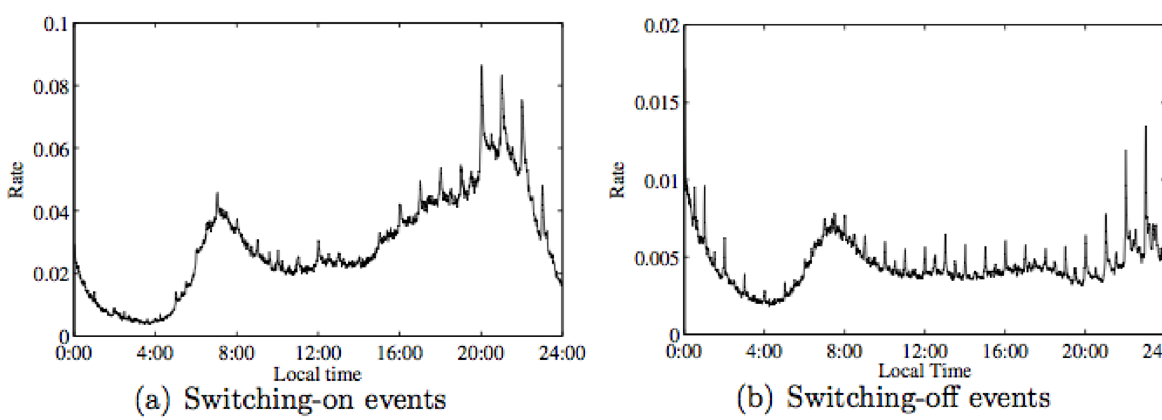

Figure 3.13: On and Off Session Activity During The Course of a Day. (extracted from [10])

3.7 Channel Switch

Channel Switch is the period of time between the tuning of a specific channel and then switching to another different one. Once again we see the same pattern of usage and activity in Figure 2.22. The main behavior is the same but we can see that this specific event has more frequent spikes along the day. The spikes period is of approximately 30 minutes, which coincides with most of the TV shows alignment. Most users switch channel when a program ends or when there is a commercial break, and that leads us to the pattern that can be seen in Figure 3.14.

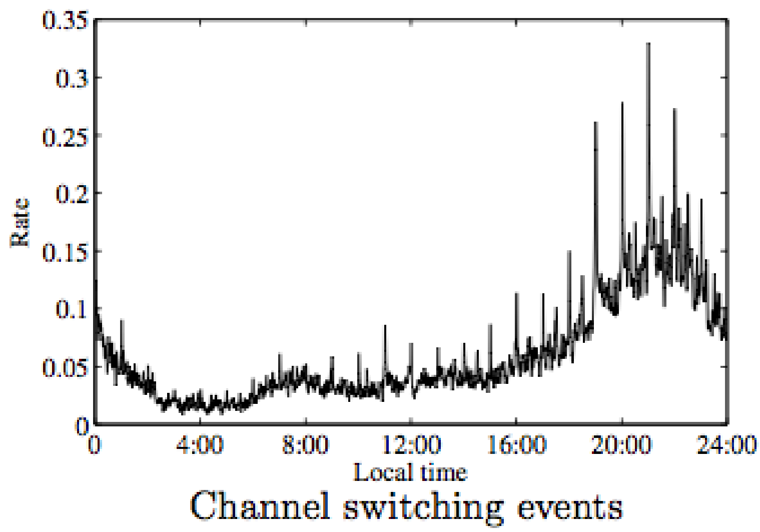

Figure 3.14: Channel Switch events (extracted from [10])

Qiu et al. have divided Channel Switching into two categories: sequential-scanning and target-switching. Sequential scanning means that the users browse through the available channels using the up/down buttons of the remote. Target switching represents the tuning of a specific channel. The total amount of Sequential switching is 54% and from those 75% are up channel switches.

3.8 Channel Popularity

Qiu et al. studied channel popularity and discovered that when it comes to popularity the top 100 channels are responsible for almost 63% of the channel viewing time(Figure 2.23). This leads us to the conclusion that the viewing habits of users are very simillar. We can also view the channel access frequency that is presented in Figure 3.15, the less skewed curve is due to the sequential channel switch events.

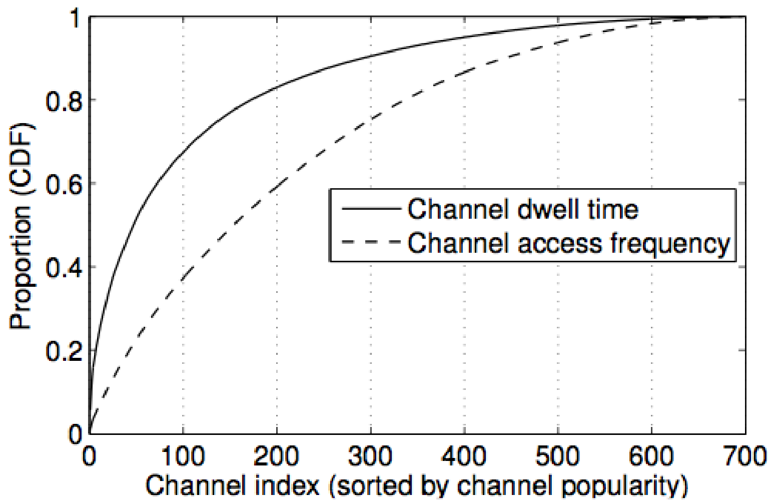

Figure 3.15: Channel Popularity (extracted from [10])

We can also discover some interesting patterns in Figure 3.16, where we see the comparison of popularity between DVR, Kids Channels and Local News. We can see that Kids and Local News Channel peaks during the morning and then loose popularity, by the other hand DVR increases popularity through the course of the day and decreases in the night.

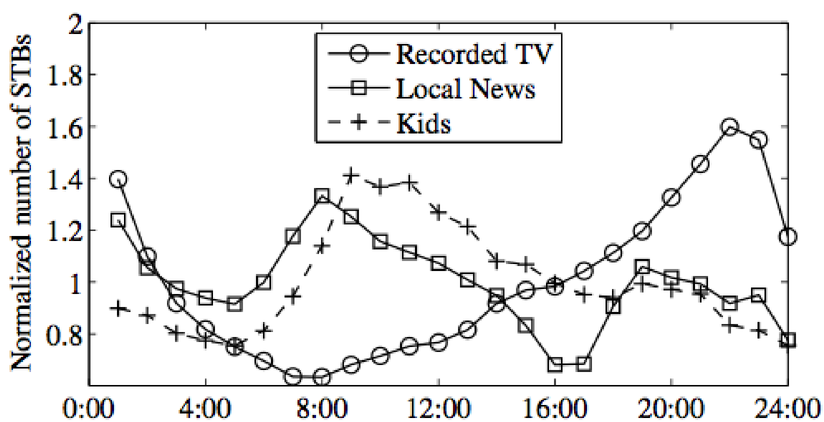

Figure 3.16: Channel Popularity Kids vs News vs DVR(extracted from [10])

This brings us tot he conclusion that people tend to view the DVR channel when they are at home, at the end of the day, because that's the period they have more available time.

3.9 Summary

In this chapter we have discussed user behavior while using IPTV services, the way they interact with the service and how we could use the STB logs to help us define this behavior.

We have come to the conclusion that there is a pattern in the service's utilization. This can help us and the ISP's provision for the peak user demands and also to study how we can make the system adapt to them.

The analysis of this data has given us some insight on how to model user behavior, the statistical data that has been gathered and studied from similar works will allow us to generate user behavior and create a life-like simulation of users activities.

Chapter 4

Cloud Computing

The evolution of network infrastructures that we have mentioned previously not only gave birth to a new and more QoS aware Internet but also paved the way to the evolution of services that are provided through the Internet.

Lately there has been a buzzword that is getting more and more attention from behalf of Information Technology (IT) Comunity, and that word is "The Cloud". It seems that The Cloud is the "silver bullet" of modern times, the technology that the IT Comunity was waiting for in order to provide anytime/anywhere services that IMS and NGN talked about. Why the cloud? The cloud has been used for long time to represent both the PSTN and the Internet, it allowed for an abstraction of all the infrastructure and interconnected systems that composed them.

So what is Cloud Computing? By the definition of the NIST [28] "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Another definition of cloud computing can be found in [29], "Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services."

Another one can be read in [30], " Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a payper-use model in which guarantees are powered by the Infrastructure Provider by means of customized SLAs"

As we can see CC is not just the software but also the hardware that is available on demand for the end user. CC can give us the illusion of infinite computing resources, they are available when the user needs them and can always be extended whenever they are not enough to do the job.

So where can we find these services? Nowhere and everywhere, the cloud is spread across several data centers that share and manage the resources so that they can bee seen/access as a single unit. In this Chapter we wish to discuss CC, it's origins and evolution, the Service models it is capable of providing and the development platforms that are available to study it.

4.1 Cloud Computing Origins

In 1961, John McCarthy stated that *"computation may someday be organized as a public utility"* [11], in the early 90's Grid computing was a major revolution in the way computational resources were shared and accessed.

Grids started by sharing resources within Virtual Organisation (VO), each VO represented an academic institution or a research project that would share it's computational resources through a set of interfaces and middleware created to manage and redistribute computational power and storage.

Grids work by providing the users with service units which can in turn be traded by access to CPU cycles and other resources. CC is thought to be a evolution of Grid computing in the sense that Grid computing paved the way to the possibility of the *"utility computing"* services by creating the paradigm of shared resources through the network and defining the tools, middleware, services and organization that compose Grid Computing.

4.2 Cloud Computing vs Grid Computing

CC and Grid Computing are two faces of the same coin, they strive to overcome the need of computational resources from behalf of both corporate and educational/research institutions. While CC is more corporate friendly and GC is more Educational prone, they both arose from the same principle, bringing computational power to the masses. They both differ in a lot of ways, mainly the security, programming model, business model, data model, applications and abstractions [11], but can interact and learn with each other in order to provide better services.

"So is Cloud Computing just a new name for Grid?" [11] we believe that yes, it shares the same principles of reducing the cost of ownership and maintenance while providing access to faster and more powerful computer resources. But also no, because GC is a more restricted use oriented paradigm, which does not scale to high numbers of machines and resources as CC. Also there is a more economical interest in CC than in GC, in order to have access to an CC environment with 1000's of computers, one just needs a credit card an access to the Internet. By the other hand to access a GC infrastructure one must be part of some kind os scientific or

research group that must share it's resources with other members of the GC community and must ask special permission to submit any kind of work to the Grid. Architecture is another major difference between GC and CC as can be seen in Figure 4.1 and 4.2.

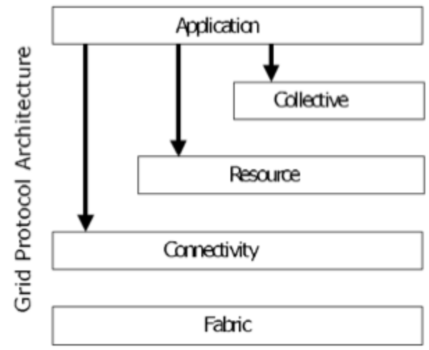

Figure 4.1: Grid Architecture.(extracted from [11])

On the Fabric Layer Grids provide access to resources wether they are computing storage or networking. On the Connectivity Layer they define the communication and authentication protocols. On the Resource Layer it defines the protocols to manage, discover, negotiate and provide operations on individual resources. On the Collective Layer it captures the interactions between resources and discovery of VO. Finally the Application Layer is made up of the applications that the users have created.

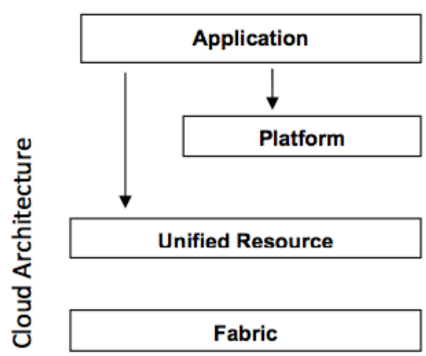

Figure 4.2: Cloud Architecture.(extracted from [11])

In CC the Fabric Layer is composed by the raw hardware, the Unified Resource Layer refers to resources that have been virtualized so they can be provided to the users through the upper layers. The Platform Layer ensures that users have a development/deployment platform through the usage of middleware. An last but no the least the Application Layer gathers all the applications that can be run on the Cloud.

In [30], we can see some of the main differences between CC and GC. Regarding resource sharing, GC defines that resources should be shared between VO's, while CC although the resources are seen as a single unit, they are not shared but virtualized. As for heterogeneity, both CC and GC are composed of heterogeneous components.

When we talk about Virtualization we can clearly see that CC uses virtual machines and resources, but GC only virtualizes it's resources as a single interface that is presented to the user. Security is also an aspect in which they both differ, CC locks access to virtualized resources to a single user, and GC requires credentials to control access to resources inside VO's.

GC offers lots of High Level Services due to it's maturity, comparatively to CC that has no such services. If we look at the architecture we can see that CC is based on virtualization of resources of any kind and GC and only use "Grid friendly" components.

Workflow is another aspect in which CC and GC differ, Grids are task oriented and need a coordinator to step through the several tasks, in the cloud the tasks are performed on demand, without the need of any kind of process management. CC uses a scalable model in which the needs of the users are always satisfied automaticly, while in GC the need for more resources requires that more nodes be added to the VO.

When we discuss usability, CC is always the winner, because GC require lots of permissions from behalf of the members of a single VO. On the other hand when we talk about standardization, GC takes the lead with it's well defined and oiled process of creation and interaction between VO's, CC still has a lot to learn in this aspect.

The payment model of both these services are very different, while GC pays a fixed rate for the usage of CPU cycles, CC uses a policy of pay-per-use of resources. Finally when it comes to QoS GC lacks the definition of a concrete policy, the usage of computational resources is made by using a task manager who's function is to submit tasks as they are presented to him, on the contrary in CC QoS is defined and applied as one of it's features.

Although GC and CC are very similar and share common interests they are also both very different, but one can't deny that both emerged from the same desire to provide computing as a public utility.

4.3 Characteristics

According to [28], CC has 5 essential characteristics, on-demand self service, broad network access, resource pooling, elasticity, measured service.

On-deman self service is the ability to provide resources and computational power without the need of any intervention within the providers os the service.

Broad network access is the possibility of accessing the data/services/apps from anywhere in the world, providing that there is a network connection wether it is done through a laptop or a smartphone.

Resource pooling means that CC has the ability to provide resources to the users from different physical locations, allocating them from where they are available, not giving a clear notion from where the resources are being accessed from.

Rapid elasticity is one of the most incredible characteristics of CC, it means that the apps or systems can grow exponentially and automatically, giving a sense of infinite resources to the end user.

Measured service can be defined as the constant monitoring and optimization of resources that are provided to the user, in order to give them the best experience available, without any hickups and system hangs.

Also on [31] we can find some characteristics that define CC, they are service oriented, loose coupling, strong fault tolerant, business model and ease use. The service oriented is achieved by masquerading the underlying structure of CC by using virtualization technologies, the end user only needs to worry about the service he wishes to purchase and nothing else. CC user can define the characteristics of the service they will acquire, in regards to storage, processing capacity and type of service, knowing that CC offers 3 types of services, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Loose coupling allows for the coexistence of several different infrastructures without any interference, virtualization allows for the sharing of resources and at the same time prevent on system's errors from reaching others, this way there is no snowball effect. CC runs in a client-server model in which the client connects loosely with the server not being dependent of data and control.

CC deployments are fault tolerant, unlike grid infrastructures a CC system does not stop and rewind to a "safe" state when and error occurs, CC has mechanisms that allow for the recovery of errors. If faults do occur in a provider of services a backups or the redundancy will come in to effect to overcome the error, if by any change the fault is between providers, the transaction will be forwarded to another provider so it can be completed.

The business model is also one of the characteristics that define CC, while on GC the main drive is for scientific knowledge, in CC the motor that drives it's deployment and development is economic interest, IT companies rollout CC services that can be used in a pay-per-use model where resources are accessed and used as if they where an utility like water, electricity or gas.

Finally we have come to ease of use, in CC, the usage of the system can be compared to the usage of a web portal where the user can select the kind of service it will purchase and the payment method he wishes to use. After having purchased the service, the user can then take advantage of it without ever having to understand how all the components come in to play to deliver the final product.

4.4 Service Models

In [30]we are presented to the concept of actors as the organizations that are involved in the deployment and development of CC infrastructures. They define Service Provider as the actor that makes services available to the service users. There is also the definition of the Infrastructure Provider as the actor that provides the infrastructure as a service. A CC provider can be a Service Provider or a Infrastructure Provider if it is only dedicated to one of these services, due to the possibility of outsourcing, it can also be both Service Provider and Infrastructure provider.

CC provides 3 models of service [28, 30], each one of them taking advantage of CC's characteristics. As we can see on Table 4.1 there are 3 types of services in CC, from top to bottom we have Software as a Service, Platform as a Service and Infrastructure as a Server.

Cloud Clients
Software as a Service
Platform as a Service
Infrastructure as a Service

Table 4.1: Service Models (adapted from Wikipedia)

Software as a Service is the possibility to use applications that are deployed on to a cloud. Users can access the required applications from anywhere in the world and anytime they want it. Software is provided like a service the users can access to in the Cloud, the consumer does not have to worry about configurations and hardware requirements because the whole process is transparent, it only needs a link, a login and a password to access his application. It can be an online alternative to a desktop solution like Gmail in case of email and Google Docs in case of Office.

Platform as a Service provides a platform to where users can deploy their applications, the end user can't control the underlying infrastructure but can create apps or deploy them providing that they have created them in a language or tool that is supported by the Cloud. Google's App Engine is one of the best examples for this kind of service, providing the tools and language to create and deploy their apps.

Infrastructure as a Service enables the user to take full control of the resources he will provide, Storage, network connections and computational resources. The user can configure all of the previous in order to create the infrastructure he needs to deploy his services in. A service like Amazon's EC2 is an example of this service model, allowing users to create and costumes their own "server".

Each one of these layers is both provider and client of the underlying one and can't function alone, Gmail could not have been deployed without a SaaS from behalf of Google, it had to be created in a PaaS environment like AppEngine that Google provided, finally both of these services would not exist if Google and IBM didn't work together to create an IaaS platform to begin with.

4.5 Deployment Models

The deployment of Clouds can be made in 4 different ways [28], Private Cloud, Community Cloud, Public Cloud and Hybrid cloud.

A Private Cloud is an infrastructure that is only accessed and controlled by a single organization, wether it is a Corporation or and University. In this kind of clouds the resources are only provided to a restricted group of users.

Comunity Cloud is simillar to the Private Cloud, the only difference is that the resources can be shared between entities that are bound together by some kind of common goal, for example Universities, Research Labs or even Technological Clusters.

Public Cloud is the one that is available to everyone, usually it is provided by corporations that sells Cloud Services.

Hybrid Cloud is a mix of the afore mentioned models, it can have a part that is Private and another that is public, or even shared between different organisattions.

4.6 Cloud Computing Standards

Companies that are developing CC haven't yet defined standards for the creation, deployment and interoperability between difference cloud vendors/providers. Currently every single CC company follows it's own implementation of the service, not worrying about standards too much, this way they can tie costumers to their own implementation of the service.

This situation has made possible CC adopters to become reluctant, due to the fear of data lock-in. If a corporation chooses to adopt a specific CC vendor and later it realizes that there is a more competitive CC provider it might wish to migrate it's data to the newly found solution. The original vendor might allow for the migration of the data if it provides a way to "extract" it from it's infrastructure by "converting" the data to the format the new provider requires. This would be the perfect situation where CC vendors would be solicit and well intentioned, but it might be the other way around. They could "force" the client to remain "chained" to their solution by not providing any way to "convert" or "extract" the data from within their solution. This kind of situation is where standards come in handy, allowing for the open and well defined interaction and exchange between CC vendor/providers.

According to [32] there are three groups of currently working on CC standards, the Cloud Computing Interoperability Forum, the Open Cloud Consortium and the DMTF Open Cloud Standards Incubator. There is also a document called the Open Cloud Manifesto where several relevant stakeholders explain why CC should implement open standards. While some believe that CC standards would be a major breakthrough for cloud interoperability, other like [33] state that there is no need of the creation of standards since all the services that are implemented within the clouds use open standards. In this document they give email in the cloud as a example, it uses SMTP[34], MIME[35], POP[36] or IMAP[37] for sending and receiving emails, despite agreeing with this point of view we believe that there are much more services that need to be implemented than email in the cloud.

Different CC vendors use different virtual machine implementations, some use XEN[32], other use KVM [32] and there are some who use VMWare, if a user wishes to port it's virtual machine to another CC provider it must be able to convert it to the used format or else he won't be able to change provider, currently there is an ongoing attempt to create the Open Virtual Machine Format (OVF) [32] standard which will help the creation of Virtual Machine Images that are vendor independent.

At the Application Programming Interface (API) level of PaaS there is AppScale[32], an open source implementation of GoogleApp's engine[38], which is aiming to become the standard.

When it comes to IaaS, Eucalyptus[32], which is an open source implementation of the Amazon EC2[32] API, is also becoming the standard, allowing for the creation and management of EC2 like/compatible virtual machines.

4.7 Cloud Computing Development Platforms

Since there is no real definition regarding the standards around CC, the different providers have come up with their own solutions to the challenges that CC brings, currently we have several major players when it comes to the Cloud. Corporations like Google, Amazon, Microsoft, IBM and Salesforce have created several products and solutions that in some cases are becoming the "de-facto standards" for CC.

Google's App Engine [39] is one of the solutions that delivers a PaaS product with the capability of automatic scaling and load balancing, it uses Java and Python for the development of software, it's popularity has led to the creation of AppScale[40], an open source implementation off it's interfaces.

Amazon with it's EC2[41] IaaS solution, probably because it was one of the first companies to develop such a product, has created such a impact on the way IaaS is planed and implemented that there is also an open source product called Eucalyptus [42] that not only mimics, but is also compatible with Amazon's EC2 deployments.

Microsoft has also created an PaaS product called Azure [43], it allows for the creation of software using its .net framework languages.

IBM, which has started defining CC together with Google, has now IaaS, PaaS and SaaS products in it's SmartCloud "suite" [44].

Salesforce [45] has taken the market by storm with the creation of their online CRM software and is becoming of the leading companies offering SaaS.

4.8 Summary

CC have come a long way since the first experimental projects made by Google and IBM, it has grown into an very attractive solution for the replacement of the internal Server/application infrastructure of corporations and has also freed the applications from the desktop prison.

Standards and interoperability within cloud environments are a long awaited step from behalf of both users and providers, this way they can offer solutions that can please every possible costumer. Several corporations have tried to foist their solutions into the market to try and take ownership of the biggest share of clients.

We have come to the conclusion that the companies that have rolled out the first CC solutions have made their solutions the future de-facto standards that are being implemented in open source solutions.

Although not being the solution to all situations, CC can achieve the needs of a great deal of costumers by providing low cost implementations or startups, anywhere/anytime availability and can gradually be expanded on demand.

CC is not the solution but can be one of the means to reach the objectives that a certain company has traced.

One of the major lacks in CC is the inexistence of standards. Each vendor tries to impose their implementation as the *de facto* standard. We believe that this is conditioning the way CC could be used by the final consumers and also by the developers who could use different CC service providers to create extraordinary applications and services.

Chapter 5

IPTV and the Cloud

IPTV is a non very far reality that has conquered the home domestic market. There are several families that enjoy the commodities that IPTV has to offer, the scheduling of recordings, the possibility of stop, rewind and fast-foward in some cases.

In order to have access to these functionalities the user must have an STB next to his TV set. This equipment is supplied by the service provider, but this comes with a cost for both the provider(purchase) and the end user(rent fee).

Today we are witnessing the appearance of new TV sets that are able to connect to the Internet, these devices provide access to web applications and allow the viewing of web content by simply plugging them to network equipment. We believe that this is the early stage of the possibility of removing one of the components needed for the reception of IPTV, the STB, which can be replaced by web application/service.

One of the technologies that is also expanding is CC, CC allows for the development of applications that are available anytime and anywhere, providing that you have access to the Internet. Besides being always available, CC apps are also elastic, increasing and decreasing the access to it's resources depending on it's needs.

With this thesis we will try to study the possibility of migrating the STB into the Cloud, meaning that all the functionalities are available in the Cloud to the end user, eliminating the need for rentings fees to be applied to the user and equipment purchasing from behalf of the service provider.

In this section we wish to talk about the way we believe these two technologies in order to create a Virtual STB that the users could access in a wide range of equipments.

5.1 Cloud Computing, the future of television.

With the evolution of technology and the lowering of hardware costs, the access to media players such as televisions, laptops, smart-phones/tablets and portable consoles has become increasingly easy.

The media content is no longer imprisoned within a specific device or network, it has become free and can be viewed and accessed everywhere. TV broadcast has increased in both quality and quantity with the rise in popularity of IPTV, allowing for the possibility of stopping, rewinding and even fast-forwarding of the broadcast.

At the present, we have the possibility of creating and sharing our own content through the use of services such as youtube or vimeo. This led to a balance shift on the content providers power. The final user is now more aware of what he wishes to see and where he wants to see it. The demand for anywhere anytime access to media and the possibility of even choosing which media has led the Content Providers and Service Providers to search for a solutions within the cloud.

The Cloud ability to expand and contract on demand is useful for Service Providers this way they do not need to over-provision for peak usage of their infrastructure [46] in case there is a spike in demand, for example in important sports events or public emergencies.

The paradigm shift of media content consumption led to the launch of new services such as Hulu, iTunes and Netflix, all these services where an eyeopener for content providers who are now considering the usage of CC as a means of storage and distribution of content.

5.2 IPTV and Cloud Computing content delivery.

According to [47] there are 3 models of IPTV currently being deployed, IMS based, NGN based and WEB based. It is expected that all of the previously mentioned models eventually will converge to a Cloud based collaborative model.

Nowadays IPTV is not the only type of service available, we also have Digital Video Broadcast Terrestrial (DVB-T) and VOD, the future of IPTV will be the merge of the above mentioned services, with the possibility of accessing then anytime and anywhere.

The increasing usage of Web based TV services like Youtube and Vimeo have changed the way users consume media and access it, in [47] there is a prediction of the number of hours that a user will spend watching "normal" TV compared to the number of hours consuming WEB based TV. The numbers indicate that by 2019 they will consume the same amount of both services with a tendency of WEB based TV overcoming "normal" TV.

This is a wakeup call for content providers, operators and infrastructure providers, they have to be capable to change the way they store and deliver content to the end user if they want to maintain them. One of the aspects is the anytime anywhere viewing "need" from behalf of the users, for this to be possible they have to be able to shift the STB paradigm and transform it into a Software Set Top Box (S-STB) that is capable of running seamlessly in a Laptop, Smartphone/Tablet or Smart-TV.

If we compare the type of service and content that is provided by the majority of both IPTV and Cable TV companies we can reach the conclusion that they do not differ a lot, they have almost the same channel list and VOD titles, in [47] they state that in France providers share 90 % of the content and we believe that it is true for most parts of the world. If all this content is the same why not store it in a "public" container and share it among providers?

This is where the Open-IPTV [47] model would come in handy, this model states exactly what we have said before, if content being provided is the same why not collect it form the same source. There should be a collective pool of content within the Cloud that would be accessed by the different providers of a specific country, this way they could offer the content to their users for a cheaper price, because not having to invest on the storage and infrastructure they could channel their financial efforts towards the development of new services and products such as the S-STB.

Sharing the same pool of resources and media leads to "security" issues that can prevent the providers from using it, but the cost saving will probably change their mindset and lead to the discussion and creation of means of securing access and distribution of contents.

5.2.1 Video On Demand and Cloud Computing

VOD is one of the services that has erupted from the massification of IPTV, the possibility of streaming the video that the user wishes to view is one of the most innovating means of content distribution, unlike video services from the Internet like youtube and vimeo, VOD is capable of sending the full video to the subscribers house and allow it to view it without any stops or intervals.

One of the greatest challenges of VOD is the storage of the content in a way that it can be streamed to the end costumer as quickly as possible, this is where CC can help. With the possibility of storage on demand and elasticity, CC can provide the storage media that is necessary for the VOD contents. CC can provide virtual servers and virtual storage that can be deployed and enabled on demand, accordingly to the capacity demands of the system[48]. This way the system can "sense" the demand on it's infrastructure and schedule the creation of new virtual VOD servers and it's storage needs.

As stated in [49], one of the solutions for VOD content distribution architecture is the Distributed Local Proxy VOD architecture. We believe that this could be a solution in which CC could help on the distribution of content, by creating virtual machines in distinct physical locations[50], closer to the users. These virtual machines would be carbon copies of each other with the same storage content an system capabilities, this way the system would not depend on a central server and even in case of failure of one of the virtual machine the others could take care of the users that where being served by it.

TV services normally broadcast all the channels at the same time, so users can switch between them without any interruption, in IPTV this is no the case.

TV channels are encapsulated inIP packets and sent through a multicast stream to all users, meaning a user that want's to see a particular channel must "tune" a specific multicast group, and by doing this it can lead to some service interruption.

Live TV and VOD is very different and the requirements on the servers are also much lower than in Live TV servers, so we can take advantage of this aspect to help us minimize the hang time between channel change in IPTV.

VOD server aren't always being used so they can be used as cache servers for Live TV streams while the user switch channels [51]. When the user want's to change to a different channel the VOD server takes over and uses it's capabilities and starts a unicast stream of the channel the user pretends, giving him the feeling of an instant channel change. Meanwhile the STB of the client leaves the previous multicast group, enters a new one and starts buffering the video stream when there STB has buffered enough video the switch is made from the VOD server to the Live TV Server.

5.2.2 Why Use Cloud Computing Simulation Tools

In this work we wanted to see if it was possible to remove the STB from the user's house and move it to the cloud so it would be accessible form within any location.

Firstly we started by trying to discover a suitable architecture for the deployment of a test base that would give us some results regarding the performance of the network and the utilization of resources. We tried to look at tools like VMWare,XEN and KVM to try and give as some sort virtual simulation environment as close as possible to the platforms that are being used by the major players in the cloud computing industry.

We came to the conclusion that the cost and the overhead of installation, configuration and testing of such an infrastructure would far exceed the time, money and the infrastructure that we had available. That's why we turned our focus into simulation tools.

A simulation can provide us with significant data so we can infer some conclusions, it saves us time that would be spent on the configuration of the test environments and it allows us to expand and implement our own scenarios.

5.3 Cloud Computing Simulation Tools

In our search for a solution to our simulator we have come across GreenCloud, GridSim and CloudSim.

GreenCloud is targeted to simulation that want to get results regarding the energy consumption of the cloud, hence it's name "Green"Cloud, despite considering that the energy consumption of a CC environment is important and should not be disregarded we did not pursue GreenCloud any further because energy consumption was not our major concern.

As for GridSim, we found that is was the genesis of CloudSim and that the simulations that it allows us to create are targeted at GC platforms, since we are not studying GC we also discarded this platform.

Finally we have CloudSim which is the evolution of the previously mentioned platform, it is targeted for the simulation of CC environments and the most mentioned platform when we made a search regarding CC simulation platforms.

GreenCloud[52] is a CC simulation tool developed by the university of Luxembourg, it is based on the well known network simulation tool NS2 [52]. It is an extension of the NS2 simulation tool that was developed by the university of Luxembourg to investigate the energy requirements of CC data centers. It helps study the behavior of applications within the CC environments, the energy models used by the components, and their respective energy demands in order to keep up with the requests made by the users.

GridSim[53] is a GC simulation environment that focus on the modeling and simulation of GC environments. These simulations are used to study the behavior of parallel computing system and evaluate scheduling algorithms that are used to manage these kind of environments.

With this simulation environment we can create heterogeneous resources that can be mixed together to create a simulation and test the availability and performance of the system while it is being accessed by multiple clients. This simulation tool was presented here because it is the genesis of CloudSim and we wanted to make a reference to it.

ClousSim[12] is a generalist CC simulation tool, it provides the possibility to create simulation scenarios to study the behavior of applications and infrastructures. It is capable of simulating, large scale CC environments, hardware and software, it is extensible and allows for the implementation of specific scenarios and customization of the platform by providing an API for developers.

5.3.1 CloudComputing Simulation Tools Comparison

After having found these CC simulation tools we had to choose one, so we had to compare them. As previously stated GridSim can not be used for our work because it only handles simulation of GC environments. With that being said we are down to two simulation tools GreenCloud and CloudSim.

Properties simulators	GridSim	GreenCloud	CloudSim
Cloud Computing Simulation	No	Yes	Yes
Hardware Simulation	Yes	Yes	Yes
Software Simulation	Yes	Yes	Yes
Network Simulation	Yes	Yes	Yes
Programming Language	Java	C++	Java
Energy Efficiency Monitoring	No	Yes	No
Latest Version/Release Date	5.2 beta /11-2010	1.0.5/???	3.0.2/11-2012
Release Year	2001	2010	2009
Nr of References	2.910	2.170	1.010

Table 5.1: Simulation Tools Overall Comparison

We can see in Table 5.1that the simulation tools have almost the same characteristics, the only thing they differ from are the possibility of simulating cloud computing environments(GridSim can't do that) and the programming language they are written in(Java for GridSim and CloudSim and C++ for GreenCloud).

Properties Simulators	GridSim	GreenCloud	CloudSim
IaaS	Yes	Yes	Yes
SaaS	No	Yes	Yes
PaaS	No	Yes	Yes

Table 5.2: Cloud Computing Simulation Comparison

In Table 5.2 we an also see some differences between the simulation tools regarding the type of services they can simulate, GridSim can only simulate IaaS while GreenCloud and CloudSim can simulate all of them.

Properties Simulators	GridSim	GreenCloud	CloudSim
Server	Yes	Yes	Yes
Virtual Machine	No	No	Yes
Multiple CPU	Yes	No	Yes
CPU Architecture	Yes	Yes	Yes
Ram	Yes	Yes	Yes
Disk Size	Yes	Yes	Yes

Table 5.3: Hardware Simulation Comparison

We wanted to view also the different hardware simulation capabilities of each tool 5.3, we found that GridSim and GreenCloud cant simulate Virtual Machines and that GreenCloud can't simulate multiple Central Processing Unit (CPU)'s, it can only simulate a server with a single CPU.

Properties Simulators	GridSim	GreenCloud	CloudSim
Router	Yes	Yes	Yes
Switch	No	Yes	Yes
Bandwidth	Yes	Yes	Yes
Latency	Yes	Yes	Yes

Table 5.4: Network Simulation Comparison

In Table 5.4 we can view the the network simulation properties of each tool, once again we can see that they are very similar regarding their simulation properties, we only see GridSim lacking the capacity of simulating Switches.

After having reviewed these tools we have decided to choose CloudSim because we wanted to be able to simulate both applications and network behavior. We also needed to be able to simulate all kinds of CC environments and hardware of different architectures and capabilities. The need for network simulation was l also a must so we could see how it would behave with a big loads of traffic. Since we had no need for Grid simulations nor energy efficiency monitoring, and also Java is the programming language we are most comfortable with we decided to go for CloudSim as our simulation tool.

5.4 IPTV in the Clouds

In this chapter we decided to talk a little bit about how we expect IPTV and CC to interact in order to provide us with and IPTV service that we can use anywhere.

Whether we are using a Smart-TV, a laptop, a CellPhone, a Tablet or Ebook reader with a network connection we would like IPTV to be available there. Our idea is to move the STB from the living room to the cloud, by removing the hardware equipment we are trying to reduce the ISP's expenses with the purchase, configuration and life-cicle management of the equipment, at the same time allowing them to offer a new service to their costumers. IPTV in the clouds is the concept of IPTV anytime and anywhere, providing you have a network connection and a equipment that is able to run the application.

5.4.1 Architecture

Our main idea, as previously stated is to move the STB from the living/bedroom and set it up as a virtual STB on the Cloud.

As it can be seen in Figure 5.1 the user is at home using his IPTV service in his SmartTV, SmartPhone, Tablet and Laptop through the Internet. This can only be achieved if we decentralize the STB and transform it into something virtual that can be accessible from everywhere.

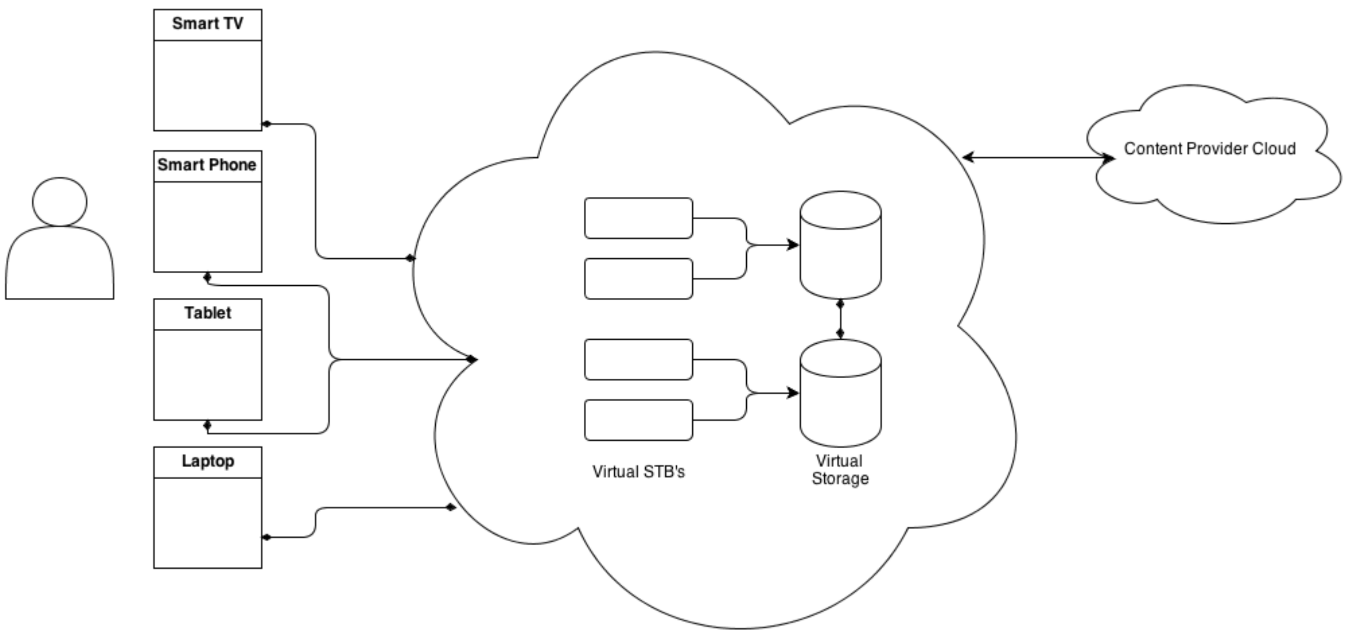

Figure 5.1: IPTV Cloud Architecture.

We have divided our architecture 3 parts, The User, The Content and The Cloud. This way we can focus on the part that we are more interested in (The Cloud) and only slightly describe the others.

5.4.2 The User

As we have already stated, The User needs the equipment to act as an interface between him and his virtual STB, he also needs a network connection from any given ISP and an account that allows him to authenticate in his virtual STB(Figure 5.2).

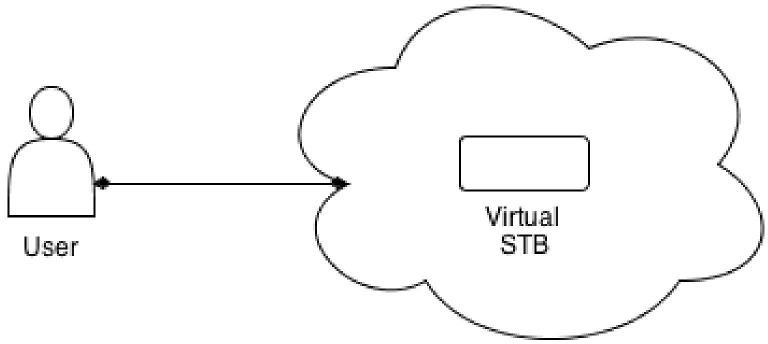

Figure 5.2: The User and the Virtual STB.

By connecting to his Virtual STB he can the view his favorite channels, recorded TV and rented movies in any equipment he desires, independently of the network connection he is using.

The User can watch IPTV through his SmartTV, in the comfort of his living room, using his home connection. He can also view his recorded show on his Tablet or SmartPhone while he commutes from home to work using a 3G or 4G connection. He can even watch a movie that he rented, at his parent's house, using their network connection to access his Virtual STB through his Tablet.

5.4.3 The content

The content is presented to the user through the Virtual STB but in order for it to reach the {STB it must be injected into The Cloud where the user's Virtual STB is placed.

As we can see in Figure 5.3, the media is injected in The Content provider's cloud, this is where the TV signal and all other media is received, encoded, stored and organized so it can be streamed/injected in The Cloud where the user's Virtual STB is located.

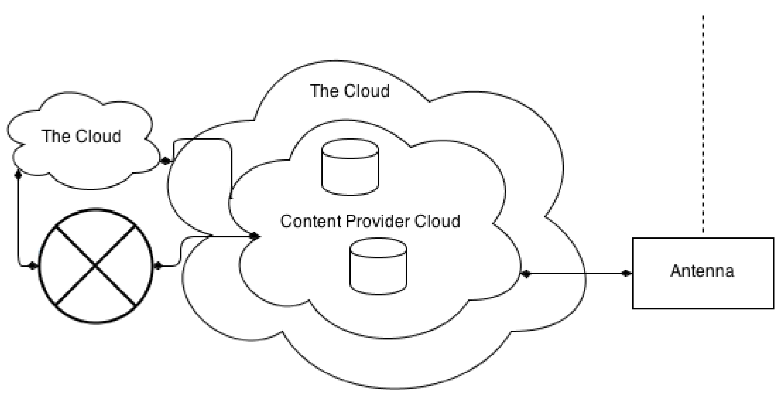

Figure 5.3: The Content.

Please notice that by dividing this part of the architecture it doesn't necessarily mean that The Content Provider's Cloud has to be in a different location than The Cloud The User has access to.

We have placed a connection through a router and through a cloud in Figure 4.3 so it is visible that The content can be sent to The Cloud in different manners. By using the same Cloud as the Virtual STB, the content can be sent directly to the equipment, if The Content is in a different Cloud Provider there is only need to make a inter-Cloud connection or use some kind of routing equipment to connect them.

5.4.4 The Cloud

The Cloud is where the user's Virtual STB is located and it is the central part of our architecture. As we have previously stated The User can access The Cloud through a variety of sources, he can use wired and wireless networks to access his Virtual STB and and view the content that he wishes.

The access to The Cloud is independent of the ISP since The User can connect to the Virtual STB using another person's network connection and still have access to his IPTV service. We can also see in Figure 5.4 what we have commented in the previous section, The Content Provider Cloud can be within the Virtual STB's Cloud or in a different Provider's Network.

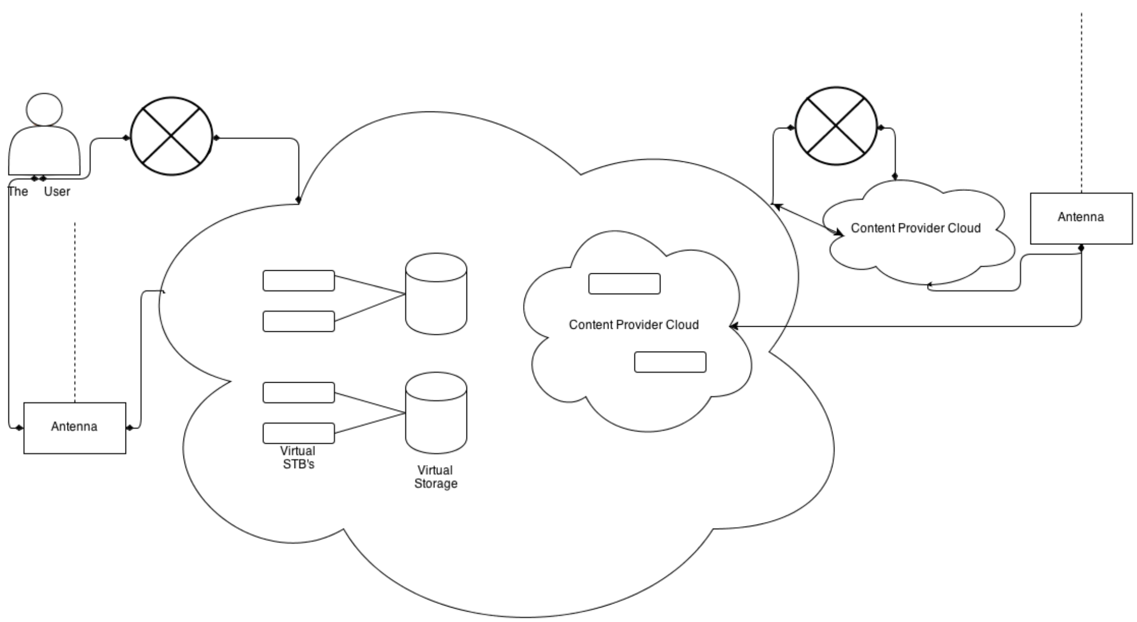

Figure 5.4: The Cloud.

We opted by moving the STB to the cloud and turn in into a virtual implementation so we could take advantage of the Cloud's possibility to grow exponentially and also be available everywhere provided that we have a network connection. This way the users are not bound to viewing IPTV only at home and have the service presented to them not as a triple play service from the ISP's. With this architecture the ISP Provider can be different from the IPTV provider and also several IPTV providers can use The same Cloud service to provide them with Virtual STB's (Figure 5.5)..

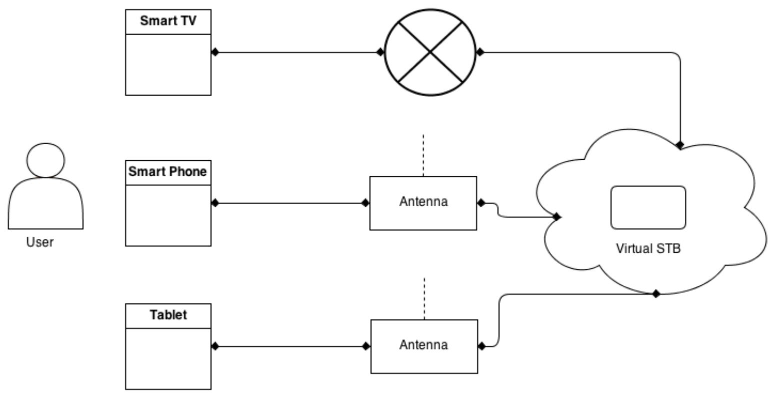

Figure 5.5: Multiple ISP's.

One of the benefits of this architecture is also that the users can take advantage of The Cloud's storage services to record TV and save rentend movies. Another service that we believe would benefit from The Cloud's storage would be the rewind or back in time TV, it could use the chunks of stored TV programs recorded by other users to reconstruct the IPTV broadcast(Figure 5.6).

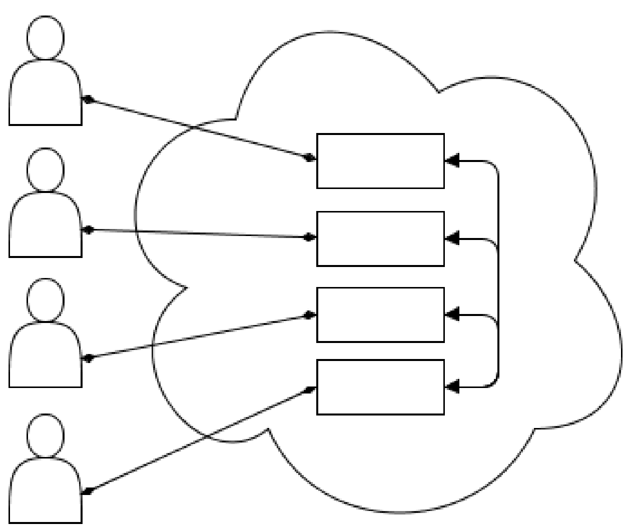

Figure 5.6: Rewind TV.

This way the users would be able to review what had already passed o TV without the need to provision special storage , this would also minimize access to TV broadcasts that have been cached in the Cloud.

5.5 Summary

In this chapter we have discussed the possibility of the interaction between IPTV and CC. We started by talking a little bit about how CC and IPTV could work together due to the evolution of the user's needs. Then we shifted our attention CC simulation tools and how they could help us simulate the environment that we needed. We have also compared some CC simulation tools (GridSim, GreenCloud and CloudSim) and chose CloudSim for the development of our work. We believe that in the cloud would be possible and we proposed an architecture that we believe could help achieve this. We have come to the conclusion that the way to go is to simulate the system before moving forward and purchase hardware and rent CC services.

Chapter 6

STB Modelling

Our work so far, has been based on the studies of other researchers, we have studied the actions of both users and STB's to identify patterns and behaviors that can be inferred from the normal usage of IPTV systems.

In this Chapter we turn our heads to the creation of diagrams that can explain how both the user and STB interact with the system. STB's are the intermediate between the user and the IPTV service, the interactions that take place between it and the user are the link we are missing in order to create a virtual STB.

We plan on studying the way users interact with the STB so we can mimic the interaction between them while creating a Virtual STB.

6.1 User Actions

We start our work with Figure 6.1 were we can view the actions that are performed by the user. He can tune a channel, browse through the PIP versions of the channels or he can issue Trick State commands.

The Trick State commands of Play, Pause, Rewind, FFW, Stop and Record are applicable on LiveTV, DVR and VOD .

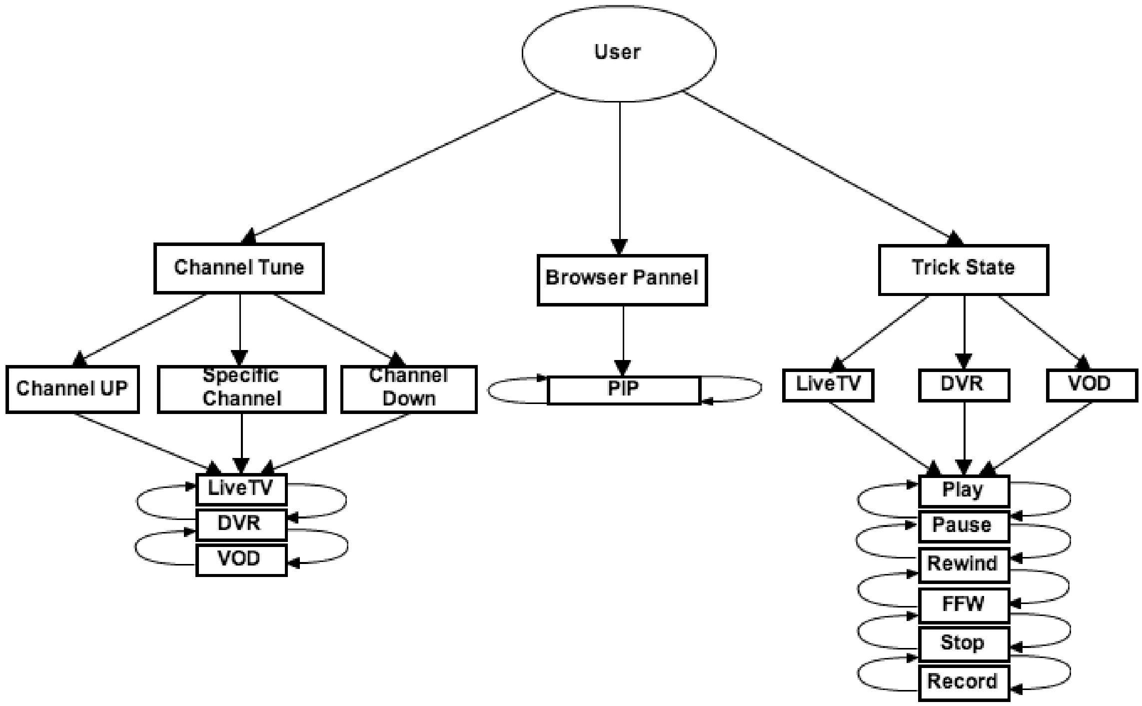

Figure 6.1: User Actions

We have also defined the actions that a STB can perform, looking at Figure 6.2 we can see that it is dedicated to the SMSR of Live TV,VOD and DVR.

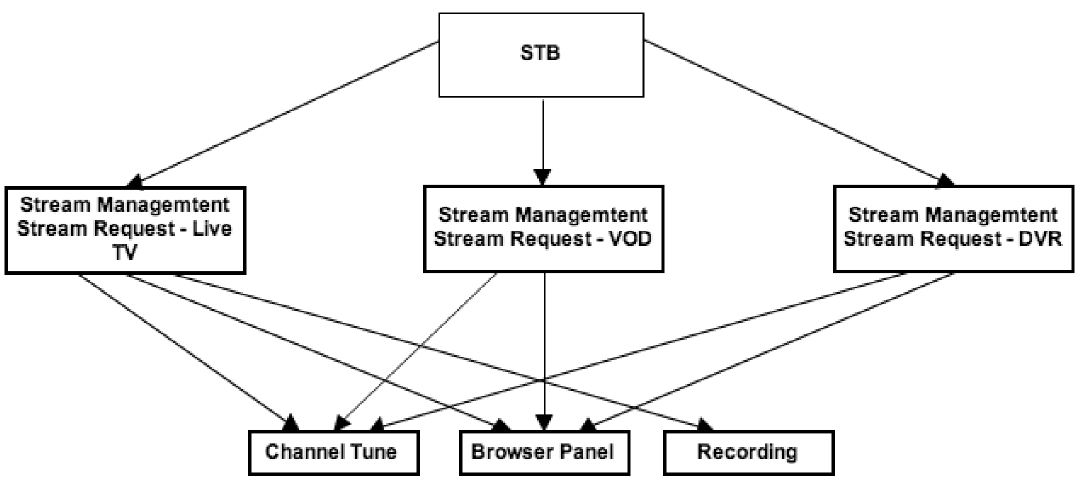

Figure 6.2: STB Actions

In the following sections we will elaborate on each of the actions that are performed by the users and describe the reaction of each action.

6.2 Channel Tune

One of the most frequent actions that a user can preform is to tune a channel(Figure 6.3), but what does that imply? By looking at Figure 30 we can see that the user viewing TV has the possibility to change the Channel Up, Down or to a Specific Channel, and by doing that he sends SMSRs.

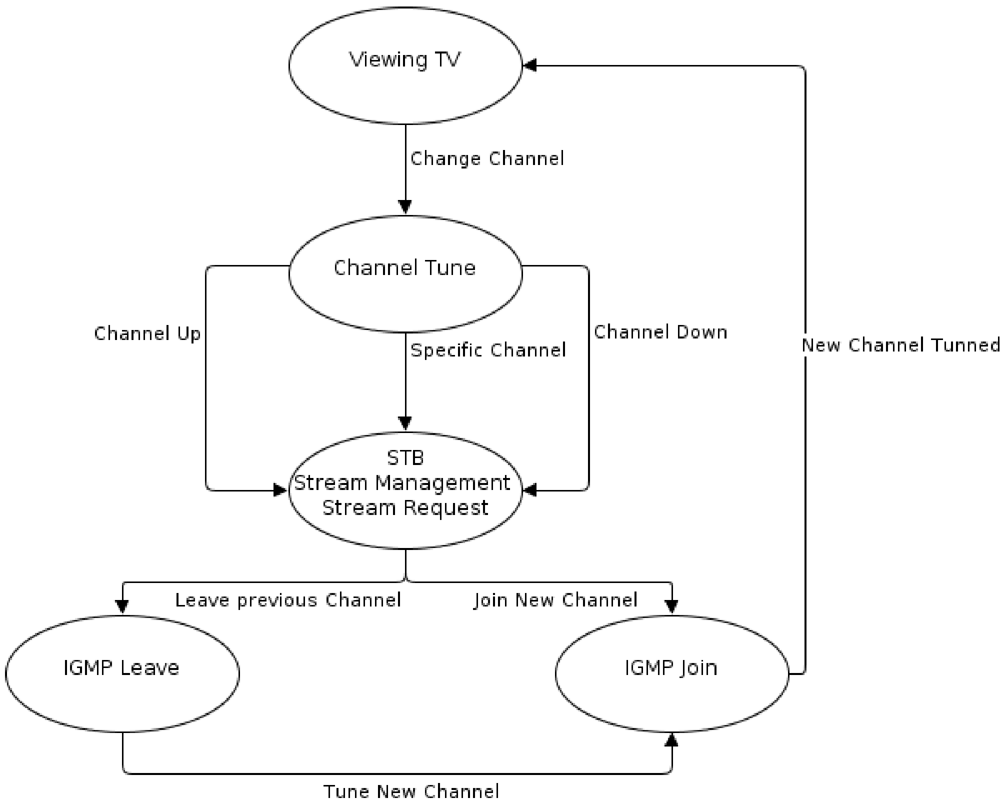

Figure 6.3: Channel Tune

The requests are then sent by the STB to the network, wether it is to change the channel up, down or tune into a specific one.

The STB will then send a IGMP Leave message to the network, informing that it no longer want's to be associated to that specific group(channel/stream). Then according to the selection of the user (Channel Up or Specific Channel) the STB will send a IGMP Join message so it can tune in/receive the selected channel/stream.

6.3 Browser Panel

While viewing TV a user has the possibility to preview the channels that are available through it's ISP. He can tune to a preview PIP version of the channel by using the GUI of the

STB so he can select a channel he wishes to tune or just to check what is going on on the other channels(Figure 6.4).

The user is watching TV and wishes to tune a PIP version of a channel, he uses the STB's GUI to access the Browser Panel, while in the Browser Panel the user can choose between previewing channels that are above or below the one he is currently watching. It might happen that the PIP channel that he wishes to view is already tuned or not, either way it is needed to send a SMSR.

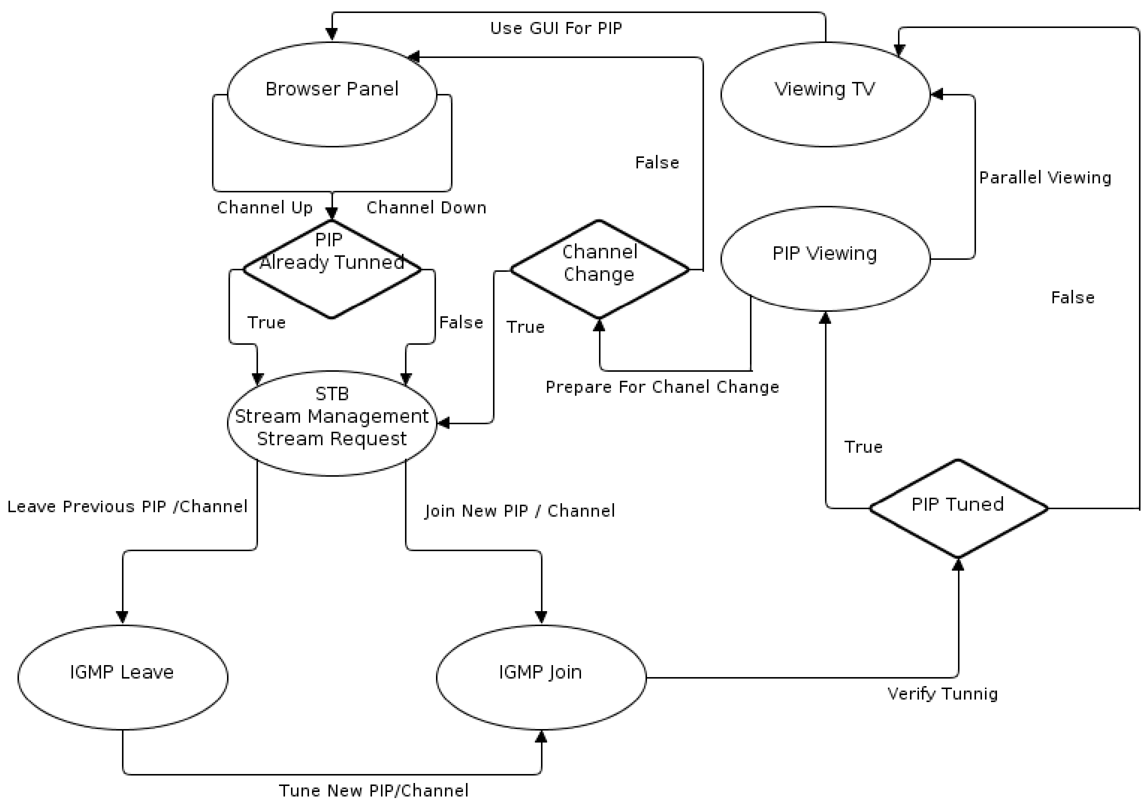

Figure 6.4: Browser Panel

The STB sends a IGMP Leave message to leave the current PIP channel (if the user wishes to view a different PIP channel) and then sends a Join message to tune to the specific PIP channel every time the user chooses a new PIP channel to preview.

After having done so, the user can select between viewing the PIP channel or quitting the Browser Panel.

When the user is viewing the desired PIP channel a Channel Change action is being prepared in the background, in case he wants to tune into that channel. The user can choose to either continue viewing Live TV and the PIP in parallel or quit the Browser Panel.

While viewing Live TV and the PIP channel the user might want to tune to the channel who's PIP he is previewing, if that is so the Channel Change Event is triggered and the process is the same as in Section 6.2, otherwise he will continue to view the channel he tuned in the first place.

6.4 Trick State

Trick State Action are special actions that the user can perform while using and IPTV system, they are divided in four groups, Play,Pause/Stop,Fast Forwar/Rewind and Record. All these actions can be preformed during Live TV, VOD and DVR , except for the Recording while using VOD or DVR. A user can not record a VOD stream to his STB internal memory because he only rented the stream and has no right to keep it to himself. The same happens to the DVR stream but for logical reasons, an STB will not record data that has already been recorded to its internal storage.

6.4.1 Play

The Play action is only logical when the previous action was the Pause or Stop action, otherwise if the user presses play while viewing a stream nothing will happen (Figure 6.5).

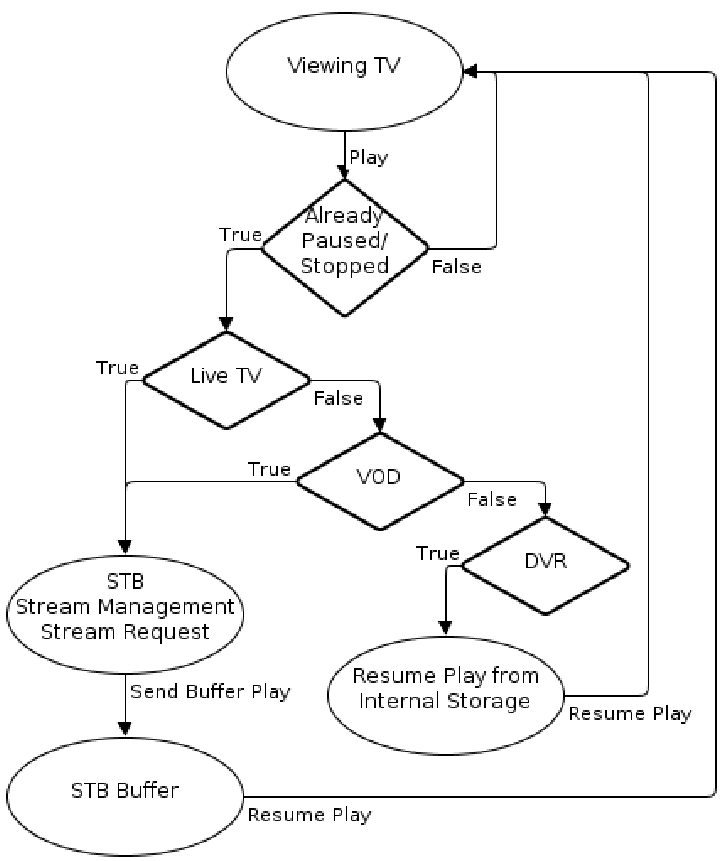

Figure 6.5: Trick State Play

If the stream is already stopped the STB has to know what it is dealing with, wether it is a Live TV, VOD or DVR stream. Depending on the type of stream the STB will trigger different actions. If the user is viewing a Live TV or a VOD stream that is already paused the STB has to send a SMSR to resume play from within the STB Buffer. If the user is viewing a DVR stream from a previously recorded program the stream is resumed from within the STB's internal storage.

6.4.2 Pause/Stop

The Pause/Stop action is similar to the Play action, the STB first needs to verify if the users is viewing Live Tv, VOD or DVR (Figure 6.6). If the user is viewing Live TV or VOD the STB's Stream Managment Stream Request sends a Buffering Request to the STB's buffer so it can store the stream while the user doesn't resume Play.

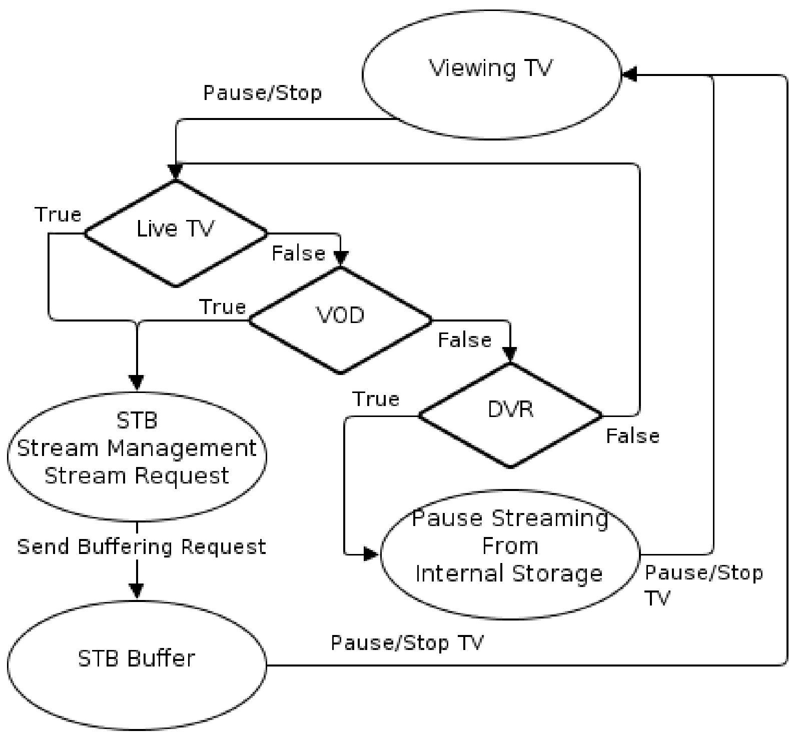

Figure 6.6: Trick State Pause/Stop

If the user is viewing a DVR stream it only pauses/stops the stream from internal storage until the user resumes play.

6.4.3 Fast Froward/Rewind

When the user sends a Fast Forward or Rewind command to the STB we first needs to see if the stream was previously Stopped or Paused and then is needs to check wether the stream is regarding Live TV, VOD or DVR (Figure 6.7). This step is needed because we can't Fast Forward Live TV if it is not Stopped or Paused, if it is not Stopped or Paused and it is not Live TV it is either VOD or DVR and for both of these Fast Forward and Rewind is possible.

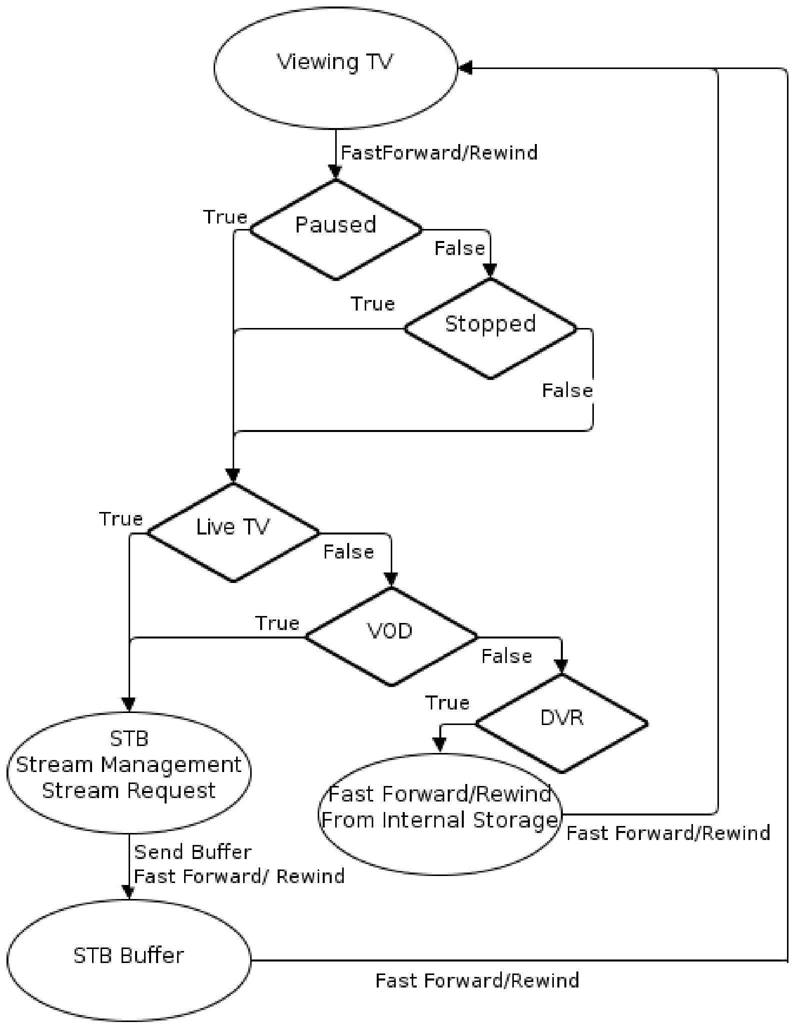

Figure 6.7: Trick State Fast Forward/Rewind

After having done the previous verifications the STB must verify if is treating Live TV (that was Stopped or Paused) VOD or DVR If the user is viewing Live TV or VOD the STB's Stream Managment Stream Request sends a Buffering Request to the STB's buffer so it can Fast Forward or Rewind the Stream from the STB Buffer. If the user is viewing DVR the STB acts on it's local Storage and Fast Fowards or Rewinds the stream.

6.4.4 Record

The last of the actions that we describe is the Record, the STB must check if the user is viewing a Live Tv, VOD or DVR stream(Figure 6.8).

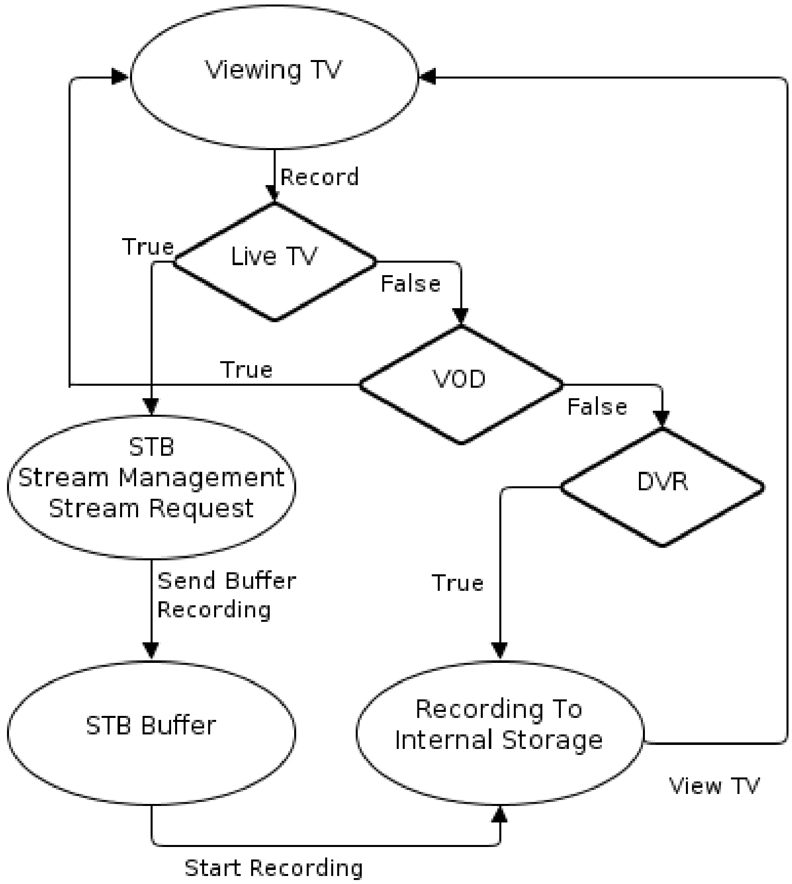

Figure 6.8: Trick State Recording

If the user is viewing a Live TV stream the STB's SMSR sends the request to store the data that is in the Buffer to the STB's internal storage. If the user is viewing a VOD stream the STB can't record it because the user doesn't have permission to store the stream to it's internal storage, so the Recording action will have no effect in this case. The case changes slightly if we are talking about DVR, the STB has permission to act on media that is locally stored but recording an already recorded stream is not logical so the Recording action will also have no effect here.

6.4.5 STB-Live Tv

While viewing live TV the user can perform the a Channel Tune, Browse the Panel for new channels and Record a specific program. We will divide the following subsections in three, so we can elaborate on these action that are performed by the user and study how the STB reacts to each one of them.

6.4.6 STB-Live TV Channel Tune

When the user wishes to shift the channel up, down or even tune a specific one he sends that request to the STB, which in turn send the request to it's SMSR(Figure 6.9). The SMSR then sends the request for the tune of a new stream to the network interface, this request is then forwarded through the network to the DSLAM/VideoHub.

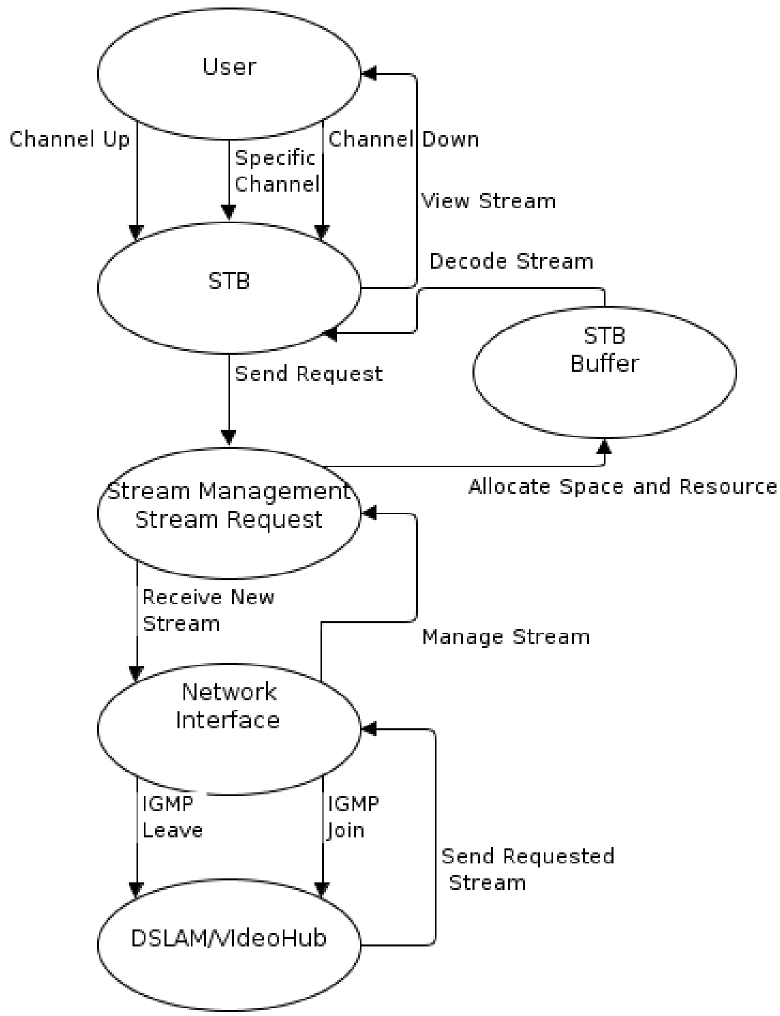

Figure 6.9: STB Live TV Channel Tune

The request is sent as a IGMP Leave and/or IGMP Join, the DSLAM/VideoHub responds to this request with the requested stream, that is sent back to the network interface. The Network Interface then forwards the stream to the SMSR with a request for management. The SMSR send a request to allocate space and resources to the STB's buffer, after having done so the buffer send a request to the STB so it can decode the stream. After having decoded the stream, the STB presents it to the user.

6.4.7 STB-Live TV Browser Panel

The user also has the possibility of browsing a PIP version of the available channels and also tune into what he is previewing, whenever he sends that command to the STB the following happens.

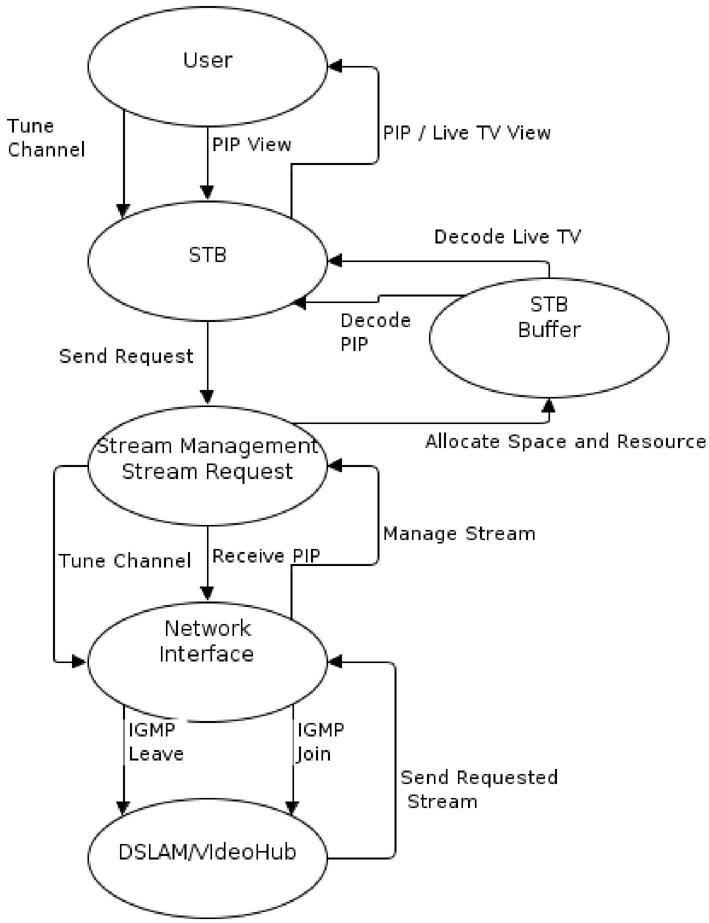

Figure 6.10: STB Live TV Browser Panel

After receiving a PIP view/Tune Channel command the STB forwards that request to the SMSR which in turn sends a Receive PIP /Tune Channel command to the network interface(Figure 6.10).

The network interface is then responsible for communicating to the DSLAM/Video Hub the need for a IGMP Leave/Join. After having received that order the DSLAM/VideoHub sends the requested stream back to the network interface.

The received stream is sent up to the SMSR to manage, it then sends an order to the STB buffer so it can allocate space and resources to receive the stream. The STB buffer then asks the STB to decode the PIP or the new live TV channel. After having decoded the requested stream the STB presents it to the user.

6.4.8 STB-Live TV Recording

When the user is viewing TV he can ask the STB to record what he is viewing(Figure 6.11). The STB first has to check what kind of stream it is already decoding, Live TV, VOD or DVR.

It has to make a logical decision here if it is already decoding a VOD or DVR stream, if it can't record it again so the result is no action from behalf of the STB. If the STB is decoding a Live TV stream it then send the request to the SMSR who requests the STB buffer for the recording of the decoded data it already has regarding this specific stream.

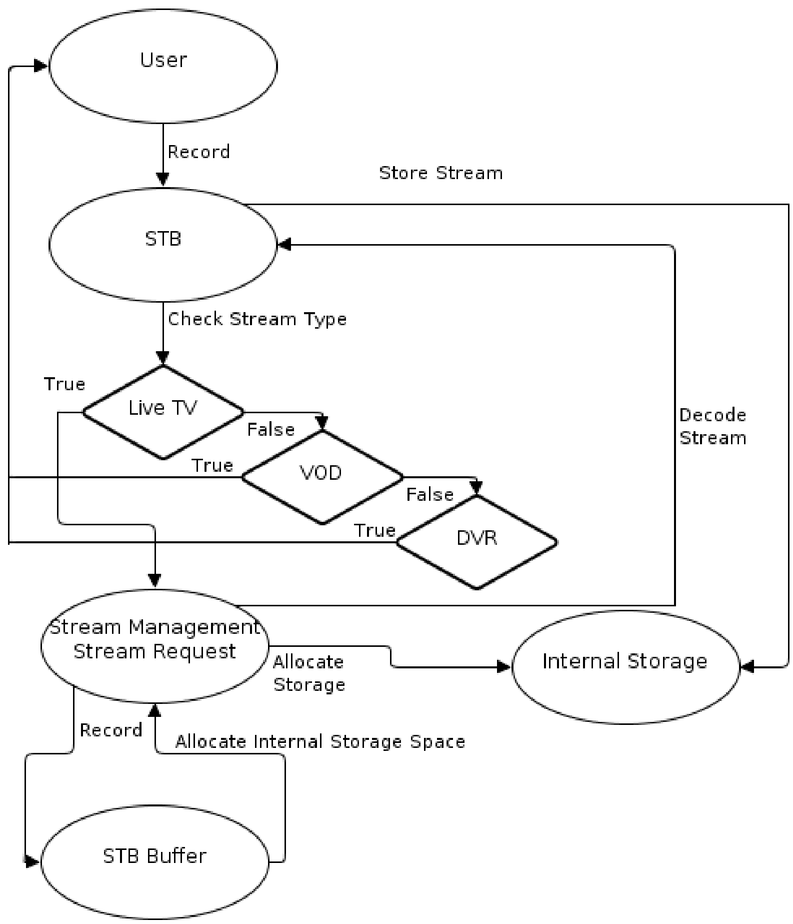

Figure 6.11: STB Live TV Recording

The STB buffer asks the SMSR to manage the allocation of internal storage for saving the data, this command is the sent to the STB internal storage so it can reserve some space for the data that is going to be sent to it.

Parallel to the command for the allocation of storage the SMSR sends the request to decode the stream to the STB which then sends the order to store the stream to the internal storage.

6.5 STB VOD/DVR

We have previously dissected the actions of the STB when it is managing commands made by the user while viewing Live TV, we turn our attention now to the events that take place when the user is viewing a VOD or a DVR stream. We have decided to separate these events because we believe there is no relation to live TV and that they relate to each other.

6.5.1 STB VOD

If a user wishes to view a VOD stream it requests the STB for a catalog of the titles that are available that request is then sent to the network interface(Figure 6.12).

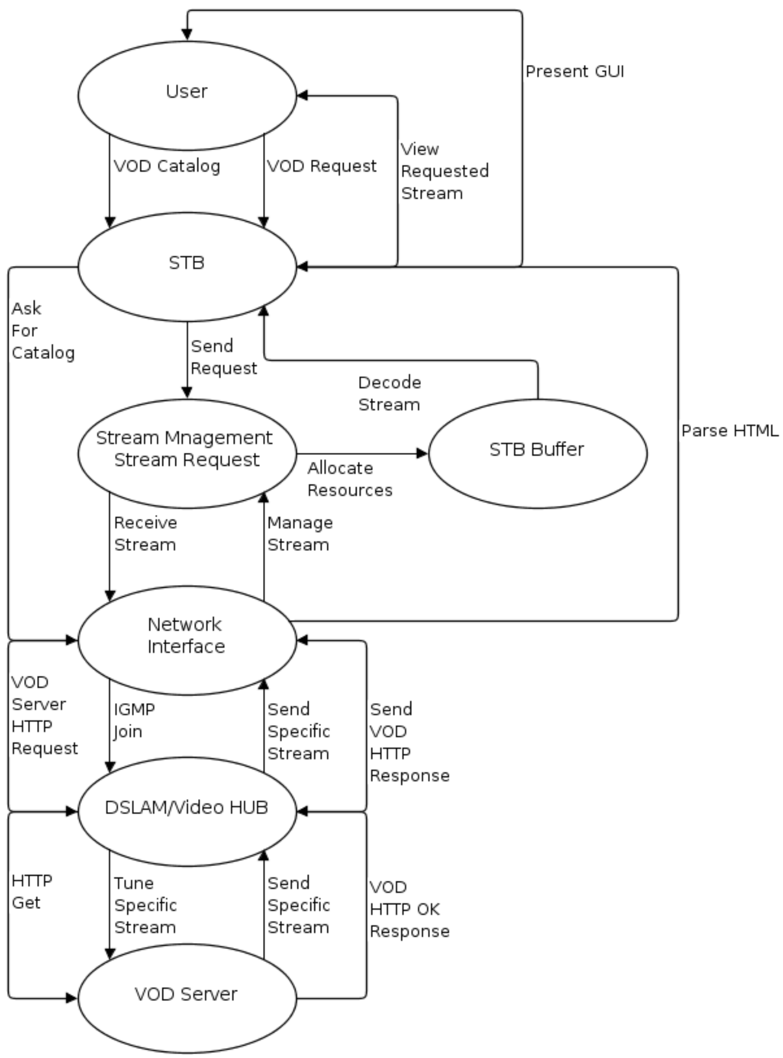

Figure 6.12: STB VOD

The network interface sends a Hyper Text Transferences Protocol (HTTP) request to the DSLAM/VideoHub that forwards it to the VOD server. The VOD server responds with HTTP ok response and some Hyper Text Markup Language (HTML) code that moved up though the DSLAM/VideoHub and network interface. The network interface requests that the STB decodes the HTML and presents if to the user in a formatted GUIso he can view what are the available Videos.

When the user selects a specific video the STB sends the request to the SMSR that in turn asks the network interface for a specific stream. The network interface sends a IGMP Join request to the DSLAM/VideoHub who forwards it to the VOD server as a tune specific stream command.

The VOD server sends the stream as unicast through the DSLAM/VideoHub to the network interface. The network interface forwards it to the SMSR that allocates space in the STB buffer for the stream and then send the decode stream command to the STB so it can present the requested stream to the user.

6.5.2 STB DVR

The user might want to view or review content that he has previously recorded and for that he uses the DVR function of it's STB(Figure 6.13).

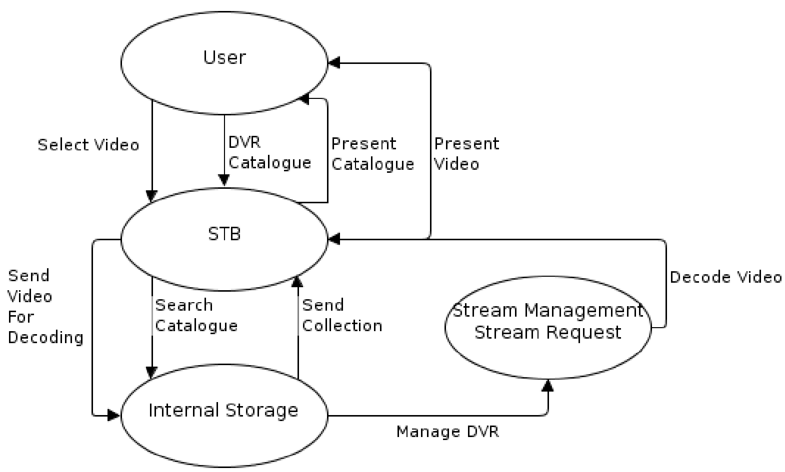

Figure 6.13: STB DVR

Comparatively to the VOD, the user needs to view a catalog of what he has previously recorded, so he sends the DVR catalog command to the STB.

The STB then searches the internal storage for the catalog, after having it gathered it send the collection to the STB so it can be presented to the user.

If the user selects a specific video the STB issues a send video for decoding command to the internal storage. The command then triggers a manage DVR command that is forwarder to the SMSR which asks the STB the decode video order. The STB then presents the video to the user.

6.6 User STB and Network Interactions

After having described the interactions between the user and the STB we now turn our heads to the interactions between the user the STB and the network.

We have decided to join all the trick state events and recording in the same diagram, we also joined live TV viewing and browser panel. We have done so because the interaction created by them are very similar, the interactions created by DVR and VOD where each also placed in a diagram.

6.6.1 Live TV and Browser Panel

We can see form Figure 6.14 that whenever the user sends a request to change the channel up, down or to tune a specific channel to the STB that request is passed on to the network through the STB. This request Generates the need for a IGMP Leave and Join in order to stop viewing the previous channel and start viewing the next one.

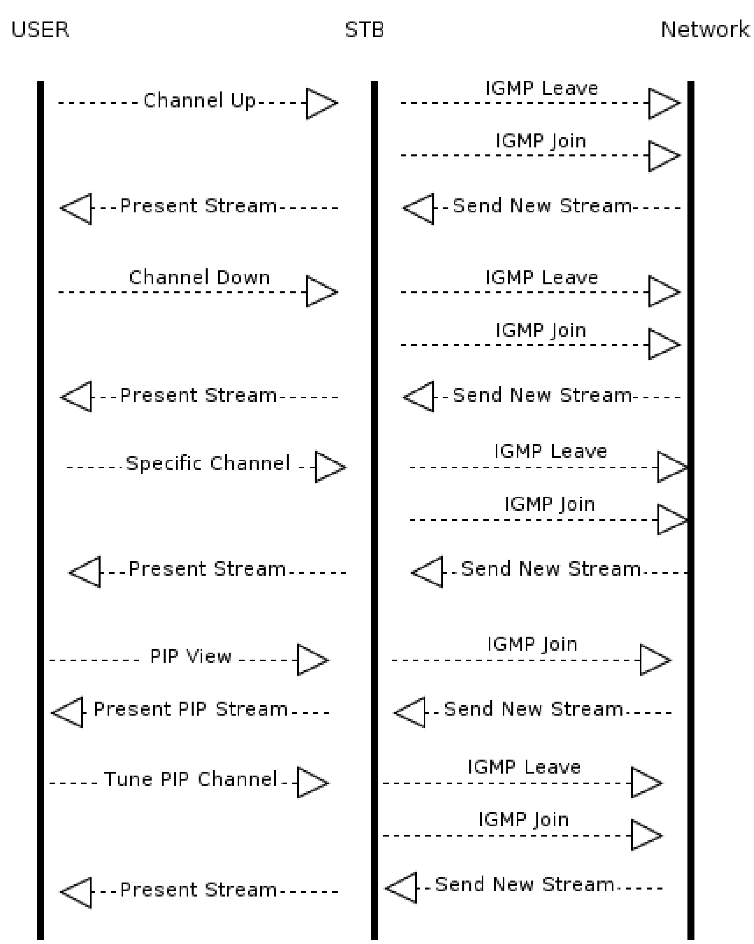

Figure 6.14: Live TV and Browser Panel Interactions

When it comes to Browsing the PIP version of channels the STB also send the request to the network so it would forward it the stream for the specific PIP version of the channel he wishes to view. If the user wishes to view that Specific channel the same procedure as TV tunning is trigered.

6.6.2 Trick State and Recording

In the diagram of Figure 6.15 we have decided to join all the trick states and recording because they are very similar, network wise. Wether we are asking the STB to pause or stop, fast forward or rewind,play or record we are always presented with the stream we have selected but in different ways.

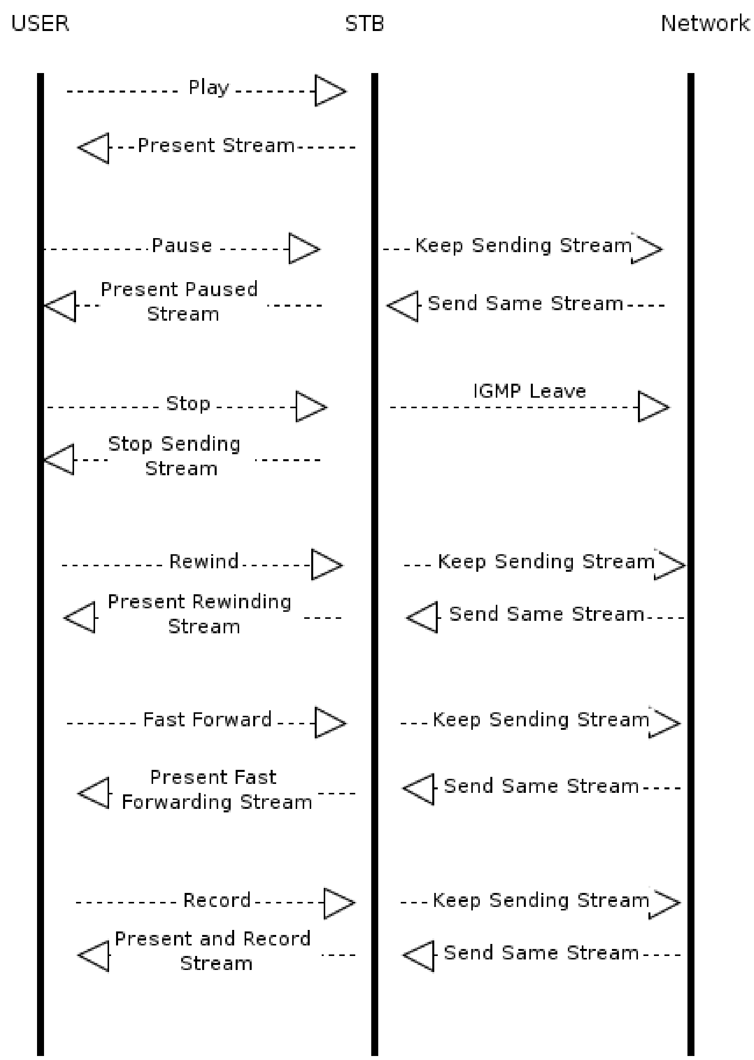

Figure 6.15: Trick State and Recording

The play action only requires the STB to continue playing a previously selected stream that was placed in one of the other trick states.

The fast forward and rewind request the STB to present the stream in a different speed, obliging it to forward that request and prepare for a burst of data to be sent by the network.

When the user selects the stop action the STB tells the network to stop sending any further data regarding that specific stream. While the pause action by the other hand forces the STB to keep on asking the network for the stream and to present a freezed stream to the user.

When asked to record the STB tells the network to keep on sending the stream, it continues to present the stream to the user but it store's it in it's memory for previous viewing.

6.6.3 Digital Video Recording

When the user wants to view a previously recorded emission he sends a request for the DVR catalog, if the request is made to the primary STB the response is the catalog. The user then chooses a specific stream that the STB send directly to him(Figure 6.16).

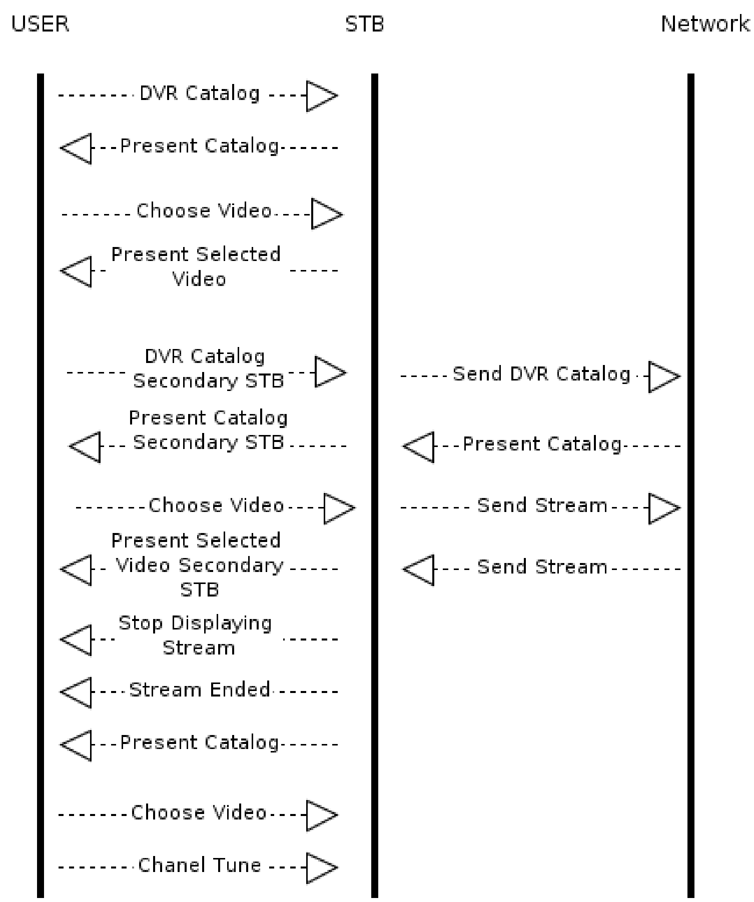

Figure 6.16: DVR

If the user is requesting the DVR catalogue from a secondary STB the command is then sent through the network to the master STB. The master STB then responds by sending the catalog though the network. After having received the data the secondary STB presents the data to the user, he then selects a specific video, the secondary STB sends that info to primary STB through the network. The primary STB then sends the selected stream though the network to the secondary STB that presents it to the user.

When the stream that was selected ends the STB tells the user that there's no more data to send and presents the catalog he then has two choices. He can choose a new video from the catalog or he can choose to tune to a different channel.

6.6.4 Video On Demand

We now turn our attention to the VOD (Figure 6.17), when the user wishes to view the VOD catalog he sends a request to the STB who forwards an HTTP request to the network. When the VOD server responds with the HTTP response the STB receives that information through the network and presents it to the user.

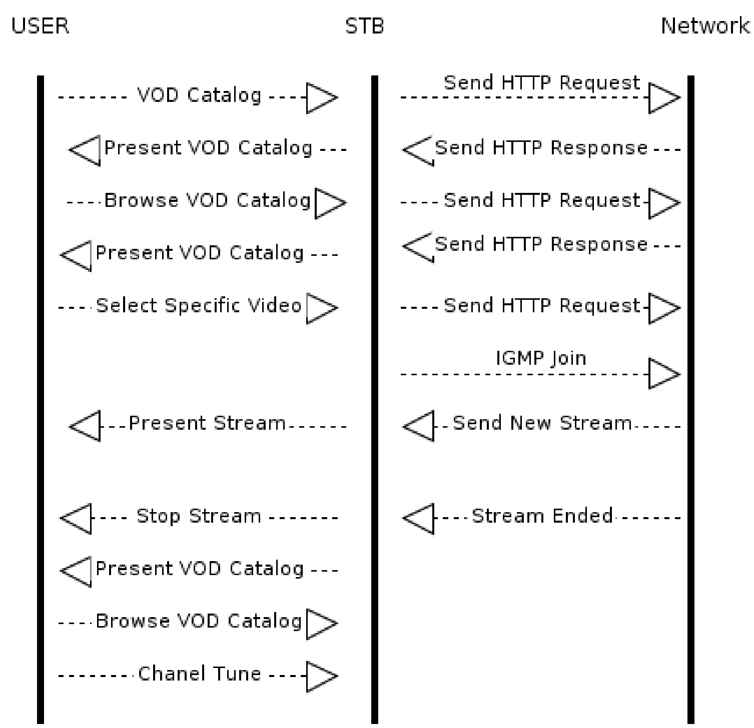

Figure 6.17: VOD

After having viewed the catalog the user then browses through it, that information is sent as HTTP traffic through the network who returns the response from the VOD server.

When the user selects a specific video the STB sends a IGMP Join message through the network to the VOD server who sends the the stream to the STB who presents it to the user.

Similar to what happens to DVR, when the stream ends the network sends the message to the STB that the stream has ended and that it will stop sending data. The STB stops the steam and presents the user with the VOD catalog he then chooses to view another video or tunes another channel.

6.6.5 ON/OFF and Standby/Resume

Finally we discuss what happens when the user turns the STB on/off, puts it in or resumes form standby(Figure 6.18) .

When the user turns the STB on the equipment sends an associate message to the network, after that he sends an IGMP join with the the first channel id from the channel list. The network send the stream that is presented to the the user by the STB.

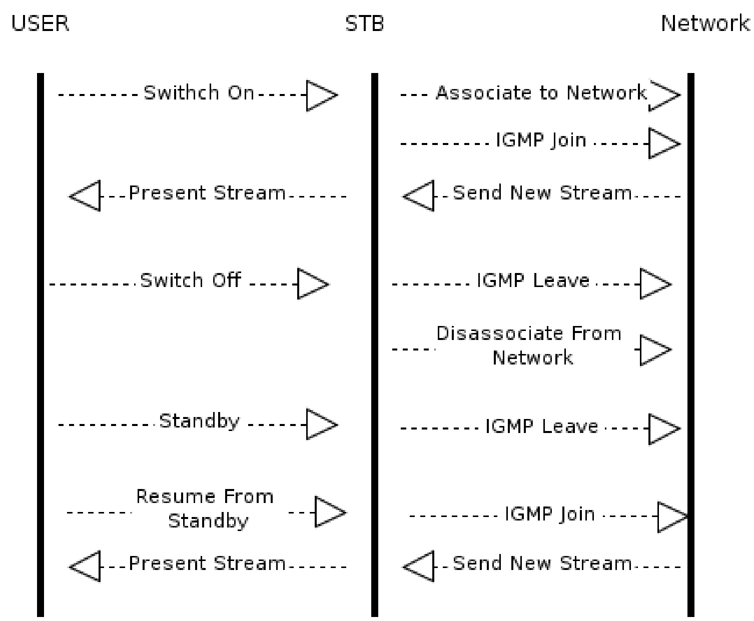

Figure 6.18: ON/OFF - Standby/Resume

When the user switches the STB off it first send an IGMP leave message to the network and then sends a message to disassociate from the network.

When the STB is on and the user sets it to standby the only thing it does is send an IGMP leave message tot he network without disassociating so it will be ready to join a channel when it is resumed.

When the user resumes a STB it sends a IGMP join message to the network so it can associate to the chanel it had previously tuned. The network send the stream that is presented to the user.

6.7 STB Logs

STB's keep track of some of the users activities in log files, action like channel change, scheduled recordings and VOD activities [9]. These logs are very useful for ISP's, allowing them to view the behavior and of both the users and the network. The information gathered by the logs can be used for the creation of reports and charts that map the usage of both network services and infrastructure.

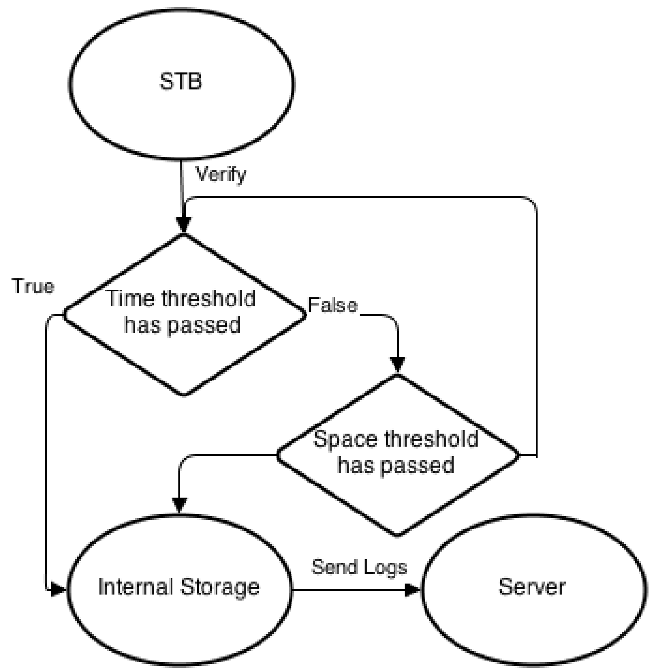

Figure 6.19: STB Logs

The STB uploads the logs to a server (Figure 6.19) when one of two this happen. If the STB hasn't sent a report to the server within a specific period of time it then send the log to the server. If the log has reached the maximum amount of space it then send the report to the server.

6.8 Summary

We believe that we have achieved the goal of modeling a STB, in this chapter we have created models that define the way both users and STB's interact with each other.

The diagrams that we have created helped us understand how the user interacts with the STB, which actions are compatible between them and those who are not.

We have also looked at the way the network and the STB interact and how are the logs, that monitor every action the user makes, sent to the ISP's main servers for processing.

These diagrams and study of behavior could prove useful to both ISP's and software developers that would like to better comprehend the interactions between the STB and the users.

Chapter 7

Experimental Results

With this work we plan to simulate how would a STB behave in a cloud environment. We plan on getting results that allow us to affirm that it is economically feasible to move the STB to the cloud.

The creation of a testbed for this work in a cloud environment would be very expensive so we chose to do it in a simulation environment. The CloudSim environment allows for the simulation of cloud environments so we chose it as our tool.

Users can interact with the STB in several ways, we call each of these events an action and for us to be able to simulate the behavior of a STB we need to create the action it should do when asked by the user.

CloudSim needs to know how to behave in such situations, so we have to map the users activities to actions it would understand, this is done by mapping the user actions to cloudlets that CloudSim can process. The cloudlets are altered in a way that they can reflect the processing of data and usage of computational resources a normal STB would use.

After having defined the action a user would do and how they would affect the STB, we need to tell the simulation how it should behave. This could be done by using logs extracted from real IPTV environments. Since it is not possible for us to obtain the real datasets we have decided to create our Log Synthesizer. This Log Synthesizer creates realistic logs that we can feed into our simulation and generate the behavior of users.

In this chapter we discuss the usage of the CloudSim platform, the changes and tweaks we had to make to the cloudlet entities in order to map user behavior. We also describe how we created the Log Synthesizer to synthesize realistic logs that are fed into our simulation environment.

Finally we get information from the simulation execution and create statistical information that will be used to confirm our desire to move the STB to the cloud.

7.1 Simulation Objectives

The ISP's investment in STB's must be quite significant, since each subscriber needs at least one of these equipments to be able to receive the TV channels he has agreed with the provider.

As we have previously stated, our work focuses on the desire to move the STB away from the living room and into the cloud. This way the provider can save money in hardware that can be channeled to development of other services and features for their clients.

With the proliferation of Smart TV's on the market we are witnessing a function shift of the TV set. It is no longer the static interface that allows us to view broadcasted TV. Nowadays TV sets have access to the Internet, have gesture and face recognition, allow us to connect external peripherals and even read CD's and DVD's.

If the TV set is capable of processing information and even decode HD video, so why not decode a stream from the Internet?

Our idea is to turn the TV into the video decoder equipment and use it as the interface to access the remaining functions of the IPTV service. All other function such as storage and management of recorded TV or VOD services are made on the cloud by the virtual STB.

All of this would allow the ISP's to save in energy and equipment costs as a result of economies of scale from having the information of a large amount of clients in the same place.

7.2 Used Simulation Platform

CloudSim is a CC simulation platform which has extended the functionalities that were available in GridSim, it allows us to create a testing scenario that simulates virtual CC infrastructures and systems.

Within CloudSim we have the possibility of creating a virtual DataCenter composed of several Hosts, each Host will then be able to run a multitude of VM's. CloudSim opens up the possibility to create a DataCenterBroker to manage the interaction between the VM's and the event's, which are called Cloudlets ,that are computed inside each VM.

CloudSim is flexible enough to let us elaborate further on the configuration of each of it's components, one can configure the costs, architecture, operating system, Hosts and other settings of a DataCenter. Each Host can be configured regarding it's ram, storage and CPU. We can create as many VM's as we wish and also setup the CPU, ram and size.

We have also mentioned that we can create events that the simulation platform will process as the user's interaction with the simulated CC environment, those events are called Cloudlets and are defined by number of CPU's needed to process it, it's length, file size, output size and utilization model.

7.2.1 CloudSim Architecture

As stated in [13], "CloudSim extends the core functionalities of GridSim, providing support for the modeling and simulation of virtualized Cloud based data center environments such as dedicated management interfaces for VMs, memory, storage, and bandwidth".

Initially CloudSim used SimJava as the engine for the simulation but it has been long since it has been discontinued to make way for new and more advanced operations.

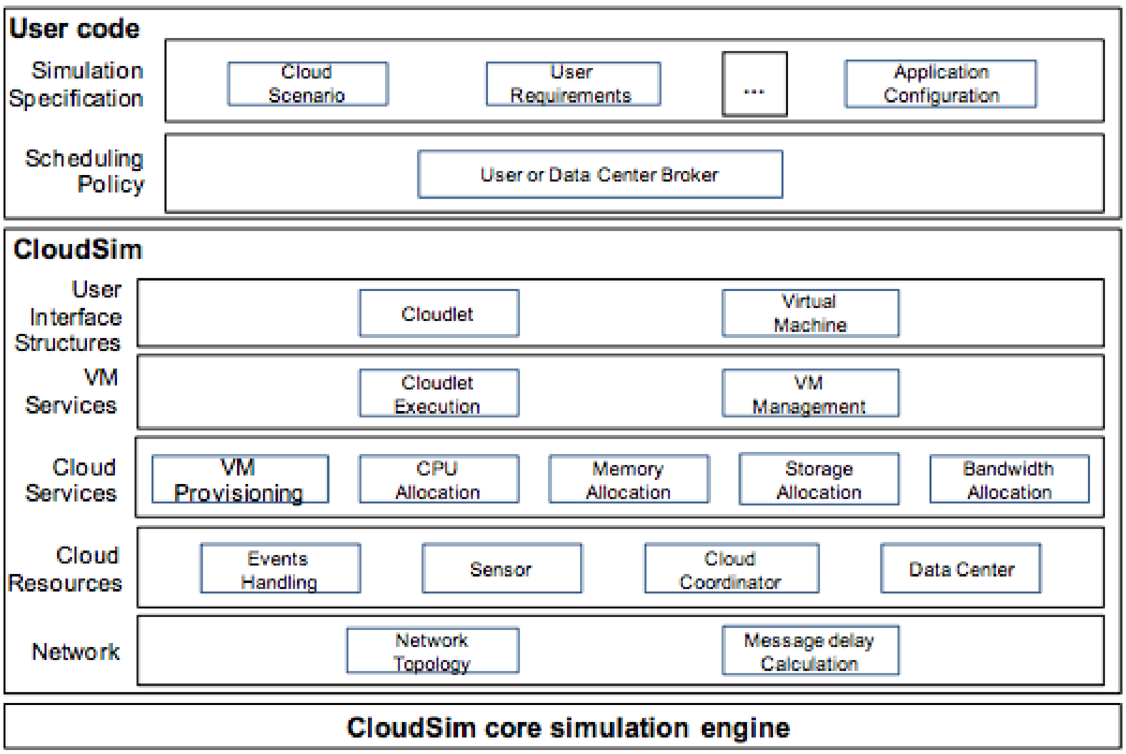

Figure 7.1: CloudSim Architecture(extracted from [12])

The CloudSim simulation layer (Figure 7.1) refers to the part where we can model a Data Center's properties, the definition of the VM's internal specification and the creation of the user's activities within the simulation. In the User Code layer we can create the scenarios for the simulation and manage the creation of the Data Center Broker and user requirements for the simulation.

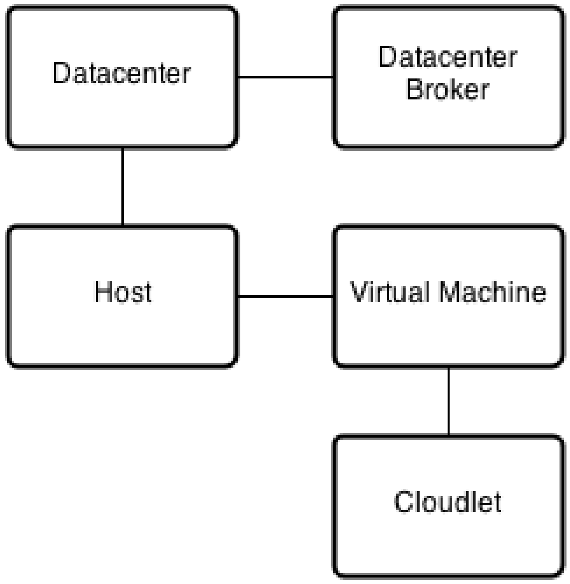

Figure 7.2: CloudSim Class Diagram (adapted from [13])

In Figure 7.2 we can see the major Classes that are used to create and define a simulation of a CC environment, as it is not the objective of this work, we will not describe all the classes that compose CloudSim in detail, we will only focus on the ones we are going to use and try to give a glimpse of what they are used for.

The CloudSim Classes

In the previous section we have showed the main classes that we use in CloudSim, in this section we want to elaborate further on each of them, explaining how they work and what are they used for. In Figure 7.3 we can see the code that helps to instantiate each one of them.

```
Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize,
utilizationModelCpu, utilizationModelram, utilizationModelBw);

Datacenter datacenter = new Datacenter(name, characteristics,
new VmAllocationPolicySimple(hostList), storageList, 0);

DatacenterCharacteristics characteristics = new DatacenterCharacteristics( arch, os,
vmm, hostList, time_zone, cost, costPerMem, costPerStorage, costPerBw);

DatacenterBroker broker = new DatacenterBroker("Broker");

Host host = new Host(hostId, new RamProvisionerSimple(ram),
 new BwProvisionerSimple(bw), storage, peList, new VmSchedulerTimeShared(peList)

Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm,
new CloudletSchedulerTimeShared());
```

Figure 7.3: CloudSim Classes Definition

Cloudlet

This class (Figure 7.3 line 1) is used to model the behavior of the applications that are run on CC environments. Here we define the computational requirement of the application to be simulated.

We have the id which is the unique identifier of this cloudlet, it is used to identify the cloudlet while it is being processed by the VMs in the DataCenter. The length is the description of the size (in MI) of the cloudlet, the pesNumber is the number of processing units needed by the Cloudlet,the fileSize is the size of the file before it was submitted to the DataCenter. The outputSize is the size of the file after it has been submitted to the DataCenter, the utilizationModelCpu/Ram/Bw are what the name implies, utilization model of the Cpu, the Ram and the Bandwidth that are required by the Cloudlet when it is being processed.

DataCenter

In this class (Figure 7.3 line 4) we simulate a Datacenter infrastructure, we define if it is composed o a single or multiple hosts and are it's hardware settings.

When we define a Datacenter we need to give it a name so we can identify it and we also need to provide it with some characteristics.

The Characteristics of a DataCenter are composed of the architecture(arch), operating system(os), the virtualization environment(vmm), a list of hosts that compose the datacenter(hostlist), the timezone it is located in(time_zone). We also have the cost of cpu utilization(cost), the cost of using memory on the DataCenter(costPerMem), the cost of using storage(costPerStorage) and finally the of bandwidth utilization (costPerBw).

The definition of the DataCenter Characteristics(Figure 7.3 line 7) is vital to the calculation of the expenses the user will have with the operation of the real datacenter and all it's cloud infrastructure.

Continuing the description of the DataCenter class we can see that we also have the VmAllocationPolicySimple(hostlist) argument, this argument is the definition of the allocation that is going to be made to the list of hosts(hostlist) to the VMs that will be run on this DataCenter. In this specific case the VmAllocationPolicySimple allocates the VM to the host with less PEs in use. StorageList is a list of storage elements used for the simulation of data being sent and received by the DataCenter, and the final argument is the scheduling interval which in this case is 0.

DataCenterBroker

The job of this class (Figure 7.3 line 10) is to manage the resource allocation of the entities involved on the simulation. IT negotiates, among other things, where the VM's should be sent within the several hosts inside the DataCenter. This class only needs an id and that's what we are doing by appending Broker and id. This id is used to identify the DatacenterBroker inside the simulation environment.

Host

Here is where we define a physical computer or storage server to be used in the simulation(Figure 7.3 line 12). We can configure how many CPU's will it have, the number of instructions per cycle, ram and storage.

In the declaration of this class we give it an id so it can be identified, we then add a RamProvisionerSimple(ram) that is used to manage the amount of ram that is allocated to the VM's by this specific Host. The same situation is managed by the BwProvisionerSimple but in this case it manages the bandwidth.

We also define the amount of storage that a Host has available, with the peList argument we define a list of CPUs with this argument we can define how many cores a Host will have available for the simulation environment.

Finally we have the VmSchedulerSpaceShared, this argument defines the allocation policy of one ore more Pes(CPUs) to a VM, in this case the VmShcedulerSpaceShared allocates one or more CPUs to a VM if we used the VmShcedulerTimeShared it would be possible for multiple VMs to share one or more CPUs.

VM

This class (Figure 7.3 line 15) models a VM that is executed within a Host. As most of the other classes this one also has an id to identify it, we need also userid so we can indicate the owner of this VM, usually we use the DataCenterBrokerId for this argument.

We also need to configure the mips that this VM is capable of computing, the pesNumber argument defines how many CPUs this machine will have. The ram, bw and size define the amount of ram, bandwidth and storage this machine will have. The vmm defines the virtual machine monitor we can use both "Xen" and "Kvm", finally we define the CloudletScheduler which can be TimeShared(severall cloudlets execute at the same time) of SpaceShared(only one cloudlet will run at a time).

7.2.2 How To Create A Simulation

Now that we have explained the major classes within CloudSim and how they are declared we will now give a brief introduction on how to create a simulation and how to alter it in order to face the problem we want to solve. The code needed to execute a simulation can be seen in Figure 7.4. First things first, we need to initialize the CloudSim environment and for that we user the code on line 1. The arguments needed are the number of user entities the CloudSim will manage, the starting time of the simulation and if there's a need for a trace.

After having initialized the simulation environment/libraries we need to create at least one DataCenter so we can run some VM's inside it. Since it is quite complicated to create a DataCenter, a function was created that simplifies this process, the function just needs a DataCenter name and all else is created programatically. the utilization of the function can be seen in line 3.

```
1  CloudSim.init(num_user,calendar,trace_flag);
2
3  DataCenter datacenter0 =  createDataCenter("Datacenter_0");
4
5  DatacenterBroker broker = createBroker();
6
7  vmlist = new ArrayList<Vm>();
8
9  Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm,
10 new CloudletSchedulerTimeShared());
11
12 vmlist.add(vm);
13
14 broker.submitVmList(vmlist);
15
16 cloudletList = new ArrayList<Cloudlet>();
17
18 Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize,
19 utilizationModel, utilizationModel, utilizationModel);
20 cloudletList.add(cloudlet);
21
22 broker.submitCloudletList(cloudletList);
23
24 CloudSim.startSimulation();
25
26 CloudSim.stopSimulation();
27
28 List<Cloudlet> newList = broker.getCloudletReceivedList();
29
30 printCloudletList(newList);
31
32 datacenter0.printDebts();
```

Figure 7.4: Simulation Howto

The creation of a DataCenter implies that we have a list of Hosts cost metrics and DatacenterCharacteristics, so the createDataCenter function takes care of all that for us.

We also need to create a DataCenterBroker to manage all the allocation of Host resources to the VMs, the instantiation of a new DataCenterBroker is similar to the creation of a DataCenter as it is done through a function which can be seen in line 5.

The next step for the creation of a simulation is the instantiation of the VMs (line 9 and 10)that are going to run inside the hosts of the DataCenter and are going to be used to compute the data required by the applications. This is done by creating a list of VM's that are then submitted to the DataCenterBroker, this way it is easier to submit multiple machines to the DataCenterBroker.We create a VM list like the one on line 7. The VMs are then sent to the broker for management, line14.

Now we need to focus on what the application behavior will be and that is achieved by the creation of cloudlets with specific needs which are then added to a list and submitted to the DataCenterBroker(line 16 to 22).

Finally we run the simulation (line 24) and get the results in the form of a list of Cloudlets (line 28) that has the status/history of each of the Cloudlets that was computed during the simulation. All that information is then printed to the users as we can see in the code from line 24 to 32.

7.3 Users Behavior Mapped To Cloudlets

For us to able to simulate the behavior of users in our simulation we need to model the user activities into cloudlets. The cloudlets in CloudSim are defined as the length in mips, the file size and output size.

In a previous chapter we have mentioned that the needed bandwidth for the streaming of a HD channel is of about 8Mbps. So by converting the bandwidth that is available from Mbps into Mb/s we get a total of 1Mb/s. We decided that we would use this result in our favor to calculate the output size and file size of our cloudlets. For example if we create an activity that lasts for one second we have a generated 1Mb of data to be stored or processed.

By using this simple calculation we where able to define how we would generate the output and file size of each cloudlet.

We have also discussed the user events in a previous chapter and in some of the cases there was a reference to the time that is normally spent by each event.

Each of the VM's that were defined in our simulation environment have the capability of 1000 mips which means that for every second there are 1000 instructions processed. Since we are working with milliseconds in the generation of our activities, by using the time an event takes to finish we were able to define the length a cloudlet would last.

For some Cloudlets, since they are mostly static, we have decided to hardcode some of the parameters, for the rest of them we have decided to synthesize their parameters.

These two variables, time and size, will accompany us through the definition of the cloudlets parameters, providing the size ad length for each one of them.

7.3.1 On and Off Cloudlet

We believe that the time to turn on and off a STB is of about 3 seconds, so while modeling these two cloudlets we defined the length it would take to complete this action to be 3000mips each, which equals to 3 seconds. Since the STB's initial state is Off, the file size is of the cloudlet is 0Mb, because nothing is being transmitted nor received, the resulting output file when it is turned on is of 3 Mb because it worked for 3 seconds at the rate of 1 Mb per second. For the modeling of the Off activity we have a state where the initial file size would be 3Mb because of the time it spent on this activity(3 seconds * 1 Mb per second) and the resulting file size would be 0Mb because after having finished nothing will be sent.

7.3.2 Channel Tune

When the users change channel they expect it to take the least amount of time possible. For this simulation we have decided that we would model the channel change so it would take only one second. With this in mind we have created a a Cloudlet with the length of 1000 mips to keep the VM working for one second. A file size of 0 because it will not process anything previously given to the cloudlet. And finally an output size of 1Mb because it was the time the cloudlet took to process and receive data. We have done so because when tuning into a channel we have no need to process previous data, but the execution of this action leads to the creation of a file of 1 Mb. While generating the activity that will give place to this cloudlet we also pass it an channel number but for now we are not using that argument since it will not increase nor decrease the performance of our simulation.

7.3.3 Session Length

We have seen in a previous chapter that the Tv viewing habits of users are very different. Each user tunes a specific channel and views TV for a limited amount of time. The users tune into a new channel for 5 minutes or less for the most part. The viewing of a channel for more than half an hour is limited to a very small amount of users.

Since it is very difficult to model all the users session lengths we have decided to choose only the intervals of 0-5, 5-10, 10-20, 20-30, and 30-60 minutes. This way we have the most common channel tuning times.

The log synthesizer we've created generates random times between the intervals that are passed as argument to the final cloudlet. The argument value is passed as miliseconds so we have to divide it by 1000 so we can get the number of seconds it took to run. Multiplying this value by one is not needed so by the end of the division we get the total output size to be defined on the cloudlet. We have also used the same calculation for the size of the file that is created inside the STB, this file is created if there is a need to review it later on.

We believe that with this range of lengths and output sizes we can get the most part of the session lengths that are described in previous chapters.

7.3.4 Off State

When the user turns the STB in to the Off State it stops all the communications between the equipment and the network, hence no data is sent from that moment on.

We believe that the time it takes for a STB to turn of is of about one second so we created a Cloudlet that would take one second to execute but does not output or process any data. The settings for this cloudlet regarding the length, file size and output size are of 1000Mips for the length, 0Mb of output and file size. This is so because there is nothing to be processed or stored during the shutdown period, the STB only turns itself off and nothing is stored.

7.3.5 Browser Panel

We believe that 5 seconds is the medium amount of time that is needed to view a PIP version of a channel and then select it. So the resulting output size will be 5Mb, the file size is 0Mb since there is no need to process anything that has been previously sent. Finally the length of the cloudlet is of 5000 because we need to pass it as millisecond value.

When we are viewing a PIP version of a channel we are already viewing a specific channel so these two action occur at the same time. In CloudSim we haven't found a way to execute two cloudlets at the same time so we have chosen to restart the action that took place before the PIP right after this cloudlet has ended it's execution.

7.3.6 TrickState

The actions performed during TrickState are pause,stop,play, rewind, ffwd and record. Some of these actions can be executed while viewing TV, using DVR and VOD and others are not applicable in some situations. The play TV action Cloudlet has already been created so we will not worry about it here.

With those ideas in mind, we have decided to create some of the actions to be specific to each utilization of the STB.

In the log synthesizer class we have defined a probable time generator that generates time periods between 0 and 60 minutes. We have used it to generate the length of a cloudlet execution argument and help with the calculation of the filesize and outputsize arguments.

Pause

Before creating our time generator we decided that the medium time for Pause and Stop would be of about 15 minutes, so the length of each cloudlet would be 900000 mips. This would lead us to the manipulation of results so to prevent this situation we used the time generated as the length argument for each one of them. By doing so we are getting more accurate results and we can reuse the output of the time generator for other purposes.

Pause TV is applicable when we are viewing TV, using DVR and VOD so we have created a pause Cloudlet for each of these activities.

The Pause TV cloudlet output size size is calculated using the generated time divided by 1000, this way we get the amount of data that is stored in the STB. This data will then be played by the user when he wishes to resume the viewing of the specific stream. As for the output size we have decided that it would be 0mb because no data will be sent or processed by the STB during the stream pause.

The VOD and DVR implies that the STB downloads/records a file in to the storage. While pausing VOD you are stopping the visualization but you are still downloading a file in the background to your STB, but for VOD you are stopping the visualization of a file that is already in yourSTB.

In the Pause VOD Cloudlet the length is given by the time that has been generated, the file size is calculated based on the time the cloudlet ran divided by 1000. The output size is 0mb because as we have previously stated when we pause the VOD activity the STB stops sending data to the user and only needs to store the data regarding the video stream on the harddrive

In the Pause DVR Cloudlet the file size is also 0mb because the activity pauses the transmission of data to the STB. The output size is of 0Mb because the data that regarding the TV show or movie that is being viewed is already in the hard drive of the STB and there is no need to generate any data.

Stop

Stop only makes sense when we are viewing DVR or VOD. We can not stop a TV broadcast, it doesn't make sense, the only logical things we can do is pause it and then resume it. With that in mind we have created a stop Cloudlet for DVR and VOD.

Initially we have decided that while viewing DVR or VOD it takes about a second to stop any of the actions so the length would be of 1000mips. After having created the time generator we decided that the time spent on the stop state is not so linear because a user might stop the viewing of a recoded TV show or rented movie for as much time as he wishes so we made some slight changes to the cloudlets.

When we pause a DVR stream we are only stoping the flow of information between the

STB and the final user, the data is already stored in the hard dive and no new files have to bee created on the disk. So the file size of the DVR stop cloudlet is of 0Mb and the output size is also 0Mb, but the length is the value given by the argument passed by the time generator.

When pausing VOD the situation is quite different because we need to store the data that is being sent to the STB so it can be viewed later. For theVOD stop activity the file size is based on the time the user stopped the streaming divided by 1000 to obtain the size of the generated file while it has been stopped. The output size is 0Mb because the STB stops sending any data to the user when it stops the VOD stream and starts saving it on the hard drive for future viewing.

Play

When we talk about the viewing of DVR and VOD we can say that the file exists within the STB since we are accessing something that has been previously recorded or downloaded.

With that in mind and knowing that most of the programs we see take about one hour, we have used the time generator to output lengths between 0 and 60 minutes, randomly, this way we can simulate the actions with more realistic times and not hardcoded ones.

The length of time the cloudlet for DVR play will run is defined by the generated random time. Since during DVR play we are sending data that has been previously stored in our STB hard drive we have to define the size of the file and the output by using the time spent on this action divided by 1000 to get the amount of data sent to the user.

For the VOD play action we have made the calculations of the length, file size and output exactly the same as the ones for DVR play. This was done so because these activities are similar to each other they both send data that is stored on the STB to the users for a determined amount of time.

Rewind

Another action that we have modeled was Rewind, we have decided to set it's length to 10 seconds because we felt it was the time it took to search for a missed scene or a not so well understood dialog in a movie or show. The mips argument for this action was defined as 10000, but after having created the time generator we changed the settings to reflect a more realistic action.

When watching live TV and we wish to rewind, what we are actually doing is getting the data that is stored on the STB hard drive or memory and search for a specific part of the stored stream. Once again we have used the generated time as the length it would take for this action to complete. The file size and output size are calculated by dividing the time spent by 1000, this way we get the size of the data that was stored on the STB for processing and the size it would have to send to the user.

Rewinding a VOD stream is exactly the same as a live TV stream because data is being sent to the STB that is stored in memory or the harddrive and then sent to the user for viewing. So the definition of a DVR rewind action is done in a similar way to the live TV rewind action, the length is dafined by the generated time and the file and output size are calculated by dividing the time spent by 1000.

When watching DVR we are viewing data the is already stored in the STB, but when we want to rewind we don't need to store anything on the memory or hard drive of the STB that is why we decided to define the file size to be 0. What we area actually doing is sending data to the user for him to choose at what point he would wish to resume play, this is why we calculated the output size as the length it would take to complete divided by 1000, the time spent in this action is generated by the random time calculator.

Fast Forward

When it comes to ffwd we have initially defined that it would take 10 seconds to reach the point we need to view if we have commercial breaks between our recorded/cached programs or trailers in our rented movie. So we defined the length of the Cloudlet to be 10000 mips. But this wouldn't be very realistic since sometimes we ffwd more than 10 seconds when viewing tv. Once again we have used the time generator to help us on the simulation.

When we want to ffwd while watching live TV we need to have some data already stored in cache or some other sort of buffer in our STB so we can ffwd. The size of the file that is going to be searched and sent to the user is defined by the amount of time, given by the time generator, we are going to spend on this action divided by 1000. This way we get the file size to be searched and the output size sent to the user.

In the case of VOD the situation is exactly the same as the previous one, we need to search the STB buffer for the data we want to send to the user. The length for this action is defined by the generated time and the output and file size are also calculated by dividing the total amount of time by 1000.

For the definition of ffwd action during DVR we configured the fileSize to 0Mb since there will be no new generation of data in the STB because all the date is already stored in its memory or hard drive. The only aspects of this action we have to define are the size of the output that is sent to the user which is calculated by the generated time divided by 1000, and the length which is the random generated time.

Record

Finally we have recording TV, which can only happen when we are viewing live TV. It makes no sense to record while viewing and already stored VOD or DVR program.

When you are recording live TV what you are actually doing is store the data that is being streamed to the user in the STB internal memory.

We have defined the length of this Cloudlet to be the same as the generated time so that we can simulate the recording of of TV according to the period that it takes place. We have also defined the fileSize and the outputSize to be the generated time divided by 1000, this is done so because the STB is sending data to the user and storing it at the same time.

7.3.7 User Activity Map

We decided to create the following table 7.1 for a better comprehension and a more visual explanation of the activities.

Activity	Length	File Size	Output Size
On	3000	0	3
Off	3000	3	0
Chanel Tune	1000	0	1
Session Length	Generated Time	Generated Time/1000	Generated Time/1000
Off State	1000	0	0
Browser Panel	5000	0	5
Pause TV	Generated Time	Generated Time/1000	0
Pause VOD	Generated Time	Generated Time/1000	0
Pause DVR	Generated Time	0	0
Stop VOD	Generated Time	Generated Time/1000	0
Stop DVR	Generated Time	0	0
Play VOD	Generated Time	Generated Time/1000	Generated Time/1000
Play DVR	Generated Time	Generated Time/1000	Generated Time/1000
Rewind TV	Generated Time	Generated Time/1000	Generated Time/1000
Rewind VOD	Generated Time	Generated Time/1000	Generated Time/1000
Rewind DVR	Generated Time	0	Generated Time/1000
FFWD TV	Generated Time	Generated Time/1000	Generated Time/1000
FFWD VOD	Generated Time	Generated Time/1000	Generated Time/1000
FFWD DVR	Generated Time	0	Generated Time/1000
Record TV	Generated Time	Generated Time/1000	Generated Time/1000

Table 7.1: Cloudlet and activity arguments

In table 7.1 we can see al the activities and the arguments that define them as cloudlets to be processed by our simulation environment. We believe that this visual explanation complements the previously written description of each one of the actitivies.

7.4 User Behavior Datasets

The first idea that we had in mind was to use some sort of User Behavior Dataset as a starting base for our simulation.

With this kind of dataset we could feed our simulation environment with real data gathered from STB logs and ISP's core routers regarding the behavior of their client while using IPTV services.

After having passed through the information on these datasets the simulation would start to execute the same actions as the real environment. This would give us some concrete data and information regarding if it would be useful or economically viable to switch the STB to the clouds.

Enable A Simulation using Datasets

In [9] and [10] the authors had access to the logs of ISP's STB's, with that information they were able to infer the behavior of users and created some of the charts and statistics that helped us during our investigation phase.

While analyzing the logs of both Silva and Qiu et al. we found equivalent file structures. Generally speaking the information within the logs is composed by a timestamp, an origin/client id, and an action. Of course there are other informations but for the sake of our work we are only interested in the ones we have mentioned.

Since we do not have access to this kind of datasets we need to create our own. Based on the statistics and discoveries of [9] and [10] we have decided to map the actions of users to cloudlets generate our own logs and feed the simulation platform with this information so we can reach some conclusions.

Activity Generator

We have decided to develop an activity generator that will use data we have gathered from other papers and will generate logs according to them.

We first need to map the user activities to Cloudlets then we need to need to create a generator that returns arrays of actions according to the probabilities we have discovered.

Finally we need to feed these arrays into our CloudSim simulation and extract some data from it.

7.5 Log Synthesizer

We need to feed our simulation with some sort of log so it will generate the same behavior as a real infrastructure.

Without the possibility to get some real user datasets we where stuck with the generation of our own logs.

We started by studying the most common actions that users execute, we then created cloudlets that would match those actions in our simulation environment.

In order for us to be able to generate logs that are very similar to the one's we find in real life we had to investigate how would a log file from an ISP. The answer was on the found in [9] since this work has a representation of the logs that they have studied.

The logs are composed of a client id, the event id, the timestamp and the attributes of the event. So if we want to simulate user logs we have to create a very similar file.

7.5.1 TimeStampGenerator

Since that not all user are connected at the same time, and they don't turn on their STB's at the same time, we had to create a a timeStampGenerator class that would gives us random generated start times of simulations.

We started by identifying the probability of the users connecting and disconnecting throughout a day, we found the probabilities that are mentioned in Table ref{tab:Starting time probabilities.

Time	Probability of Occurrence
00:00-04:00	5%
04:00-08:00	20%
08:00-12:00	10%
12:00-16:00	10%
16:00-20:00	25%
20:00-24:00	30%

Table 7.2: Starting time probabilities

With these values mapped we were able to return a starting time of each user's logs.

With the timestamp argument given the class we define a 24 hours period in which the user should turn on his STB, we then use this 24 hours period to generate a starting and end point for our simulation. We have decided to generate also a 24 hour period simulation so we can view how will our cloud environment behave during the course of a full day's work.

The resulting start and end timestamp are then saved to a file that will be accessed by a future class.

7.5.2 LogSynthesizer

Now that we have a file with the start and end timestamps for each user activities we can start generating it's actions.

We decided to create a LogSynthesizer that would be capable of using the statistics and data gathered from previous studies and return some behavior log files we could feed our simulation with.

Our SynthesizeLogs class recieves a endtime, a userid and a starttime arguments. This class has several methods that generate the actions that will be used to create a user log.

We started the creation of our Log Synthesizer by creating a Probable Cloudlet Generator, this class creates cloudlets based on their probability. But this alone was not sufficient because, besides having to create cloudlets based on their probabilities, we also have to take into account that not all user actions are compatible.

We needed to make sure that the actions the Probable Cloudlet Generator class was creating were compatible, for this to happen we created Validator classes for each of the user actions. These validators take into account the actions that have been created previously and only generate events that are logical from a IPTV utilization point of view.

We have defined five main actions as the starting point to our event generation, these actions are On, Live TV, PIP, Trick State, Off and Turn Off. For each of these actions we have defined a probability of them taking place, we have used the information we gathered from [9] , [10] and also our own experience as users of IPTV services. In Table 7.3 we can see the probabilities we have defined.

User Action	Probability of Occurrence
On	10%
Live TV	35%
PIP	5%
Trick State	40%
Off	5%
Turn Off	5%

Table 7.3: Major User Actions Probabilites

Some of these main actions can derivate to other actions, for example Live TV can derive to Live TV, DVR or VOD. We have define those derivations and probabilities in Table 7.4

Live TV	Trick State
Live TV 35% DVR 20% VOD 10%	Live TV 70% DVR 20% VOD 10%

Table 7.4: Derivated Actions and Probabilites

Each one of these actions and sub-actions have sub-sub-actions of their own with probabilities that are defined in the following tables

We start by defining the Table 7.5 for Live TV probabilities, we have only defined the length up to 60 minutes because the percentage for sessions with higher amount of time is very small

Live TV Length	Probability
0-5 minutes	70%
5-10 minutes	20%
10-20 minutes	7%
20-20 minutes	7%
30-60 minutes	6%

Table 7.5: Channel Tune Probabilities

After having defined the probabilities for each of the main actions and the sub-actions we now need to define the probabilities o each sub-sub-action. We will define these probabilites in the following three tables, one for Trick State Live TV(Table 7.6), another for Trick State VOD(Table 7.7)and finally Trick State DVR(Table ref{tab:Trick State DVR Probabilities).

Live TV Action	Probability
Channel Tune	60%
Pause	5%
Rewind	10%
FFWD	5%
Record	20%

Table 7.6: Trick State Live TV Probabilities

VOD Action	Probability
Play	80%
Pause	10%
Rewind	2,5%
Stop	5%
FFWD	2,5%

Table 7.7: Trick State VOD Probabilities

DVR Action	Probability
Play	80%
Pause	10%
Rewind	2,5%
Stop	5%
FFWD	2,5%

Table 7.8: Trick State DVR Probabilities

All of the above stated probabilities helped us define and create our log generator.

We created the ProbableCloudlet method that takes care of the generation of the main actions according to the probabilities show on Table 7.3

We also have created a method for the generation of the sub-actions, we called it SatististicalCloudlet, it takes a main action as it's sole argument and then according to that sub-action it chooses which sub-sub-action it will generate.

The generation of the sub-sub-actions is made using methods that also were created according to the probabilities defined in each of the previous tables.

After having passed through the generation of a action according to the probabilities of the main,sub and sub-sub-actions our generator will then pass it's result to a main action validator. The validator will then verify if the action is acceptable, and by acceptable we mean compatible with previous actions.

We can not accept actions that are not compatible, for example tune a channel if the STB is turned off or record while viewing a VOD stream. To prevent this kind of situations we have created a validator for each of the possible actions, each validator takes into account the previous action to make sure that inconsistent actions do not happen, keeping this simulation as real as possible.

Some of the actions like ON, OFF and STANDBY we have defined a static execution time, but for others we needed to generate some sort of execution time. we did this by creating two methods. The first one generates time between 0 and 60 minutes, we have decided that 60 minutes would be the longest time spent on any action. The second method generates random time between specific intervals to be used for the generation of TV viewing times, we have defined the intervals as the ones we can see on Table 7.5.

Since some actions have already the length hardcoded we only need to create the probableTime class to generate the random time an actions would last between it's two arguments, minimum and maximum time. As stated before we have defined the maximum time an action would last is of about 60 minutes.

For those actions that have static time defined we have hardcoded the argument that is passed to the log file, as for the remaining ones the time that is generated is passed as an argument that will then be used during the generation of cloudlets.

7.5.3 Log Synthesizer Finite State Machine

For a better understanding of how our LogSyntehsizer works we decided to create a diagram where we define all the states that are possible and the probabilities associated to them.

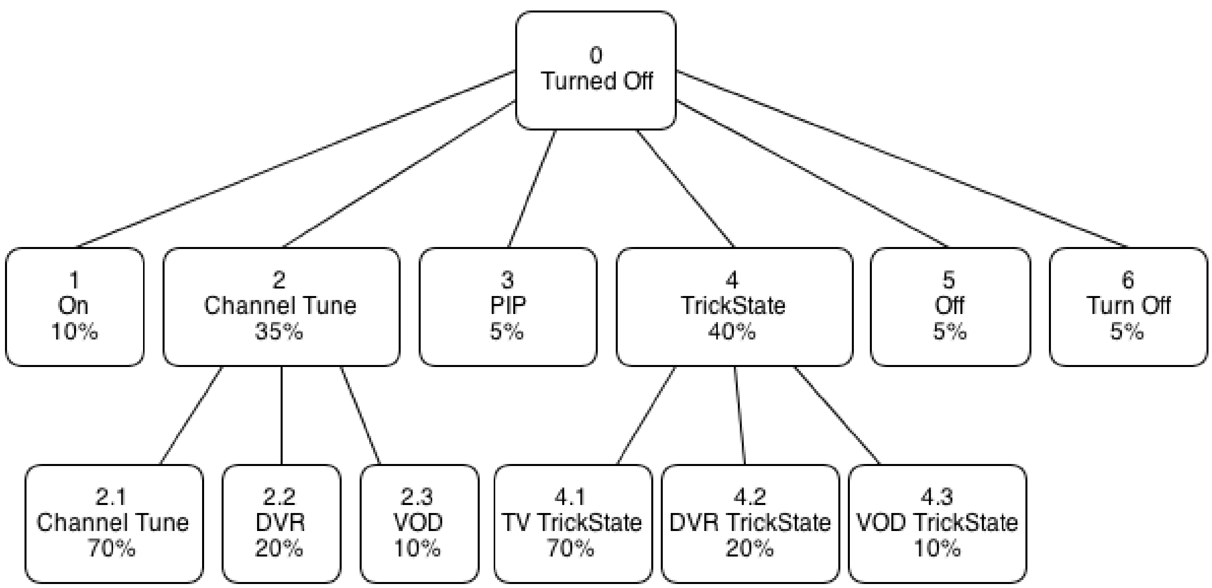

Figure 7.5: Main Activities

In Figure 7.5 we can see the main activities and probability of occurrence that our Log synthesizer can create and the first level of sub activities that are possible. We can notice that some of the activities only have on level of complexity, this is because there are no possible sub-activities that can derive from them.

The activity in the first level is the first state of our machine and the starting point of our Log Synthesizer. Every time our Synthesizer needs to start creating activities it starts with the initial state, then he looks to the probabilities in the first level generates a random probability and goes to the next level, passing through all the possible actions until he reaches a finite state. After having reached a finite state and a new action is required the Synthesizer will evaluate the generated probability and the process starts all over again.

As previously stated, we have defined the probabilities for each of the activities based on the meta-analysis of the data collected in [9] and [10].

Since the several possible states of the log synthesizer are quite a few we couldn't create a single diagram, so we decided to divide them in sub-diagrams for better comprehension.

Channel Tune

In Figure 7.6 we can se the first of our sub diagrams, we can see that the probabilities of the sub-activities are not defined only by the main activities, they also have their sub-probabilities of occurrence. Here we can see that each activity can generate sub-activities and sub-sub activities.

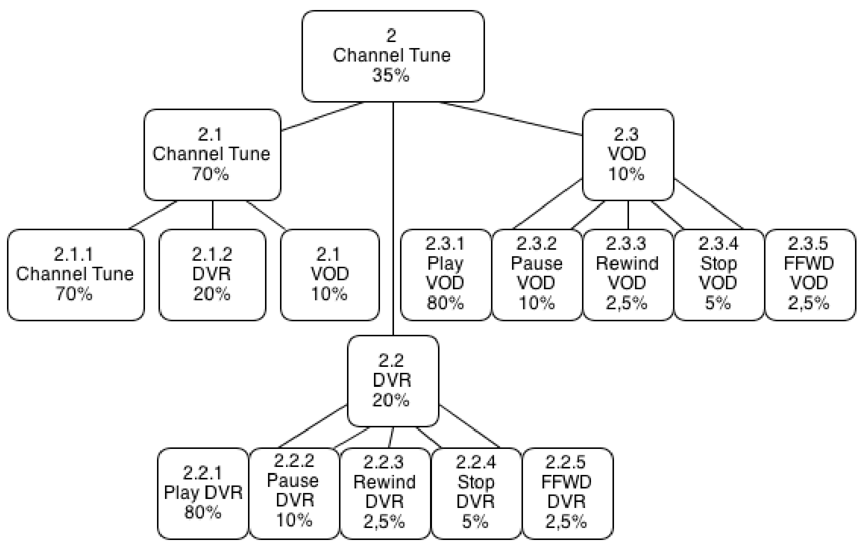

Figure 7.6: Sub-Activities Chanel Tune

From the initial state of Channel tune can derive to Channel Tune(2.1), DVR(2.2) or VOD(2.3) and each one of the previous action can derive into further action of their own.

In Figure 7.6 we can see the probability of occurrence of Channel Tune(2) and the possible actions that might occur. But this is not the end of channel tune activities because we are still missing the possible actions and probabilities regarding the time spent viewing the selected channel.

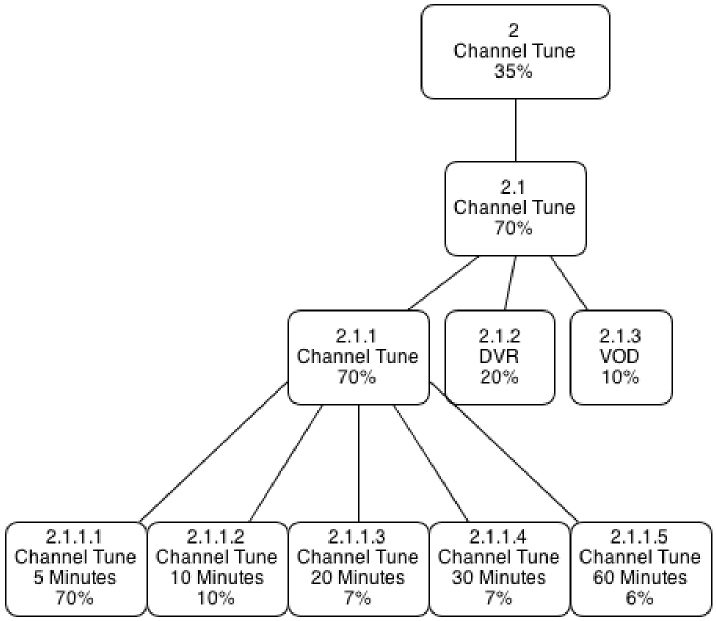

Figure 7.7: Chanel Tune Time

In Figure 7.7 we can see the time a user can spend after tuning into a channel and probability of its occurrence. These time values were the ones we discovered as the most frequent ones for an IPTV session length. These are the possible activities for Channel Tune(2.1.1), but there are other possibilities of occurrence after Channel Tune (2.1) has been selected and those activities and their sub-activities can be found in Figure 7.8.

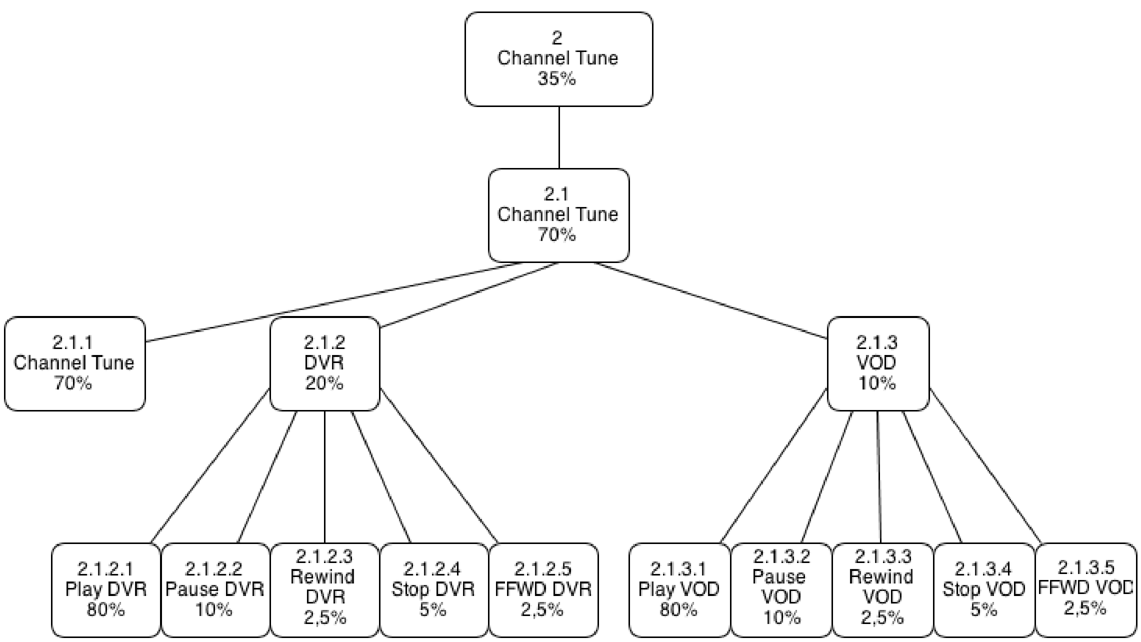

Figure 7.8: Chanel Tune Trick State

We can see the probability of our Synthesizer generating actions that involve Trick State action on DVR(2.1.2) and VOD(2.1.3) after having done a Channel Tune(2.1). It might seem quite odd to have placed these activities here but there is an explanation the DVR(2.1.2) and VOD(2.1.3) activities might occur after a Channel Tune(2) and a Trick State(4).

Trick State

In Figure 7.9 we can see our third sub-diagram, this one regards the activities that derive from the main TrickState Activity(4). There are quite a few similarities between this figure and the previous one, that is due to the fact that some activities regarding Trick State(4) of DVR and VOD have a probability to occur after a Channel Tune(2) and a Trick State(4) activity is generated.

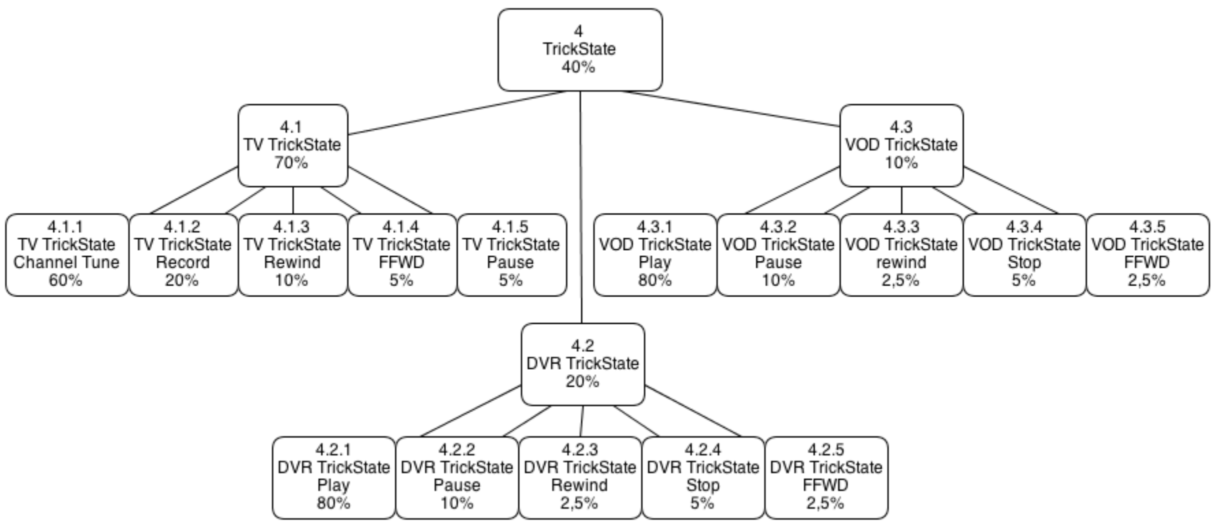

Figure 7.9: Sub-Activities TrickState

Taking a look at the sub-activities and sub-sub-activities of TrickState(4) in Figure 7.9 we can see that there are actions that are similar to the ones on Chanel Tune(2.1) but there are other possibilities that have not been referred yet like the TV Trick State(4.1)

We believe that by creating these diagrams we can shed some light on the way our Log Synthesizer works. We collected the data from [9] and [10], then by analyzing the charts they produced we arrived to the probabilities shown in the previous section. After having having collected the probabilities we started to study how the activities are related to each other and what are the probabilities of each one of them occurring after the previous one. This way we were able to create a general model for the activities and their antecedence and precedence. The creation of a full diagram would be possible, but in the end it would turn out to be very confusing for the readers of this work, that is why we decided to divide them in smaller ones.

It was our hope to achieve an intelligible explanation of our Log Synthesizer and we believe that we have achieved this with our Log Synthesizer Finite State Machine diagrams.

7.6 Generated Log Example

With the interaction of the components we developed we were able to generate a file full of events that simulate the utilization of an IPTV product from behalf of the users(Figure 7.10).

```
2013-04-16 05:02:41,84,STBY,2,3000.0,121
2013-04-16 05:02:44,84,ON,1,3000.0,122
2013-04-16 05:02:48,47,CHANNELTUNE,3,4.0,203
2013-04-16 05:02:51,55,VODPLAY,17,186000.0,146
2013-04-16 05:03:01,53,VODPLAY,17,234000.0,222
2013-04-16 05:03:02,67,CHANNELTUNE,3,2.0,211
2013-04-16 05:03:03,89,DVRPLAY,18,204000.0,271
2013-04-16 05:03:11,90,VODPLAY,17,54000.0,106
2013-04-16 05:03:12,78,ON,1,3000.0,194
2013-04-16 05:03:14,5,DVRPLAY,18,246000.0,79
2013-04-16 05:03:14,96,VODPLAY,17,78000.0,176
2013-04-16 05:03:19,53,DVRPLAY,18,18000.0,223
2013-04-16 05:03:20,3,CHANNELTUNE,3,1.0,332
2013-04-16 05:03:20,67,CHANNELTUNE,3,2.0,212
```

Figure 7.10: Synthesized Log Sample

We can clearly see the time stamp followed by the user id, after that we have the activity type in human readable form and after that the activity type id, following we have the time it takes for that activity to run in miliseconds and finally the activity sequence from the original user generated file.

These files are then parsed and sorted by date into a single log file, this file is then given to our CloudSim simulation who will use it to generate the specific cloudlets each line describes. The simulation will then execute and return the data regarding the cost of processing, the time spent for each action and the datacenter energy expenses.

7.7 Simulation Results

While executing our simulation we want to extract data that will allow us to prove that it is economically viable to move the STB to the cloud.

We need to get information regarding the processing cost of each of the actions users execute, we also need to view the cost of energy and the way both of these costs grow with the increase of the number of users.

Statistical information like the evolution of processing cost per hour, the number of activities per hour and users per hour are also useful so we can provide the cloud with sufficient resources for the processing of the user actions.

There are other kinds of costs like bandwidth and storage utilization, cost of energy and processing power per user, but those will not be discussed in our work.

To get those costs we had to simulate the behavior of the STB in the cloud. We started by simulating 25 user during the period of 24 hours, then we increased the number of users in the simulation in increments of 25 until we reached 100 users.

For each increment of 25 users we simulated a period of 24 hours 30 times, the data that resulted from those simulations was then gathered and the results we will discuss are the ones we have reached.

7.7.1 Datacenter Configuration and Cost

Since we are using CloudSim, we had to follow some of the configurations that the developers have chosen, if we hadn't done so we would have to make severe changes to CloudSim source code.

In Cloudsim, as we have previously mentioned, a Datacenter is composed of a list of Hosts with some cpu, memory requirements , storage and bandwidth size.

We have chosen to create our Hosts with 5 cpus(1000 mips capacity each),8192 mb of ram and 1TB of disk. Our Datacenter will comprised of several Hosts according to our needs.

We have defined that our VM's, the representation of the STB, would have a requirement of 1000 mips, so each one of the hosts can run 5 STB's, this can be changed further to evaluate the performance of the Datacenter under load.

The definition of the prices applied for each of the variables that account for the expenses we face at the end of each simulation were calculated using a study performed by TechRepublic[54], a site dedicated to IT Management.

We believe that is a credible study, since it is made by collecting data from the industry, and gives us a chance to view the prices that are actually being practiced in the real world.

The prices we have come up with are a mean price calculated using the data from [54], so the cost of storage is of 0.00015 and the cost per bandwidth is 0.00014.

The cost of using a Vm in CloudSim is made by the following calculations(7.1):

$$Cv = (r * Cr) + (s * Cs) \tag{7.1}$$

Where Cv is the virtual machine cost, r is the amount of ram, Cr is the cost of ram, s is the size of the virtual machine and Cs is the cost of storage.

The cost of using the DataCenter to process Cloudlets is made by using the following calculations(7.2):

$$Cc = (Cb * Cf) + (Cb * Co) \tag{7.2}$$

Where Cc is the cloudlet cost, Cb is the cost of bandwidth, Cf is the cloudlet file size and Co is the cloudlet output size.

As we can see we have two cost, one fixed(Cv, each VM will cost the same price because it's settings are fixed) and a variable one(Cc because each cloudlet has different requirements), the sum of these two costs (7.3) gives us total amount that has to be spent to run our environment in the Cloud.

$$Tc = Cv + Cc \tag{7.3}$$

With this data in hand we will be using our simulation and present some result regarding what would be the cost of deploying STB's in the clouds.

7.7.2 User Action Processing Cost and time

The cost of each activity increases with the time it uses the cpu for processing and the size of the outputted file. As we can see from the first two charts (Figure 7.11 and 7.12) the processing cost of the user activities increases according to the time each one lasts. Activities like viewing turning the TV on and Off (activity 1 and 2 from the charts) have low cpu needs so their cost is very low. On the other hand actions like the VOD and DVR (activity 17 and 18 from the charts) demand more from the cloud as it would be expected, since both of them tend to take a lot of time and demand more processing power than the simple viewing of a stream.

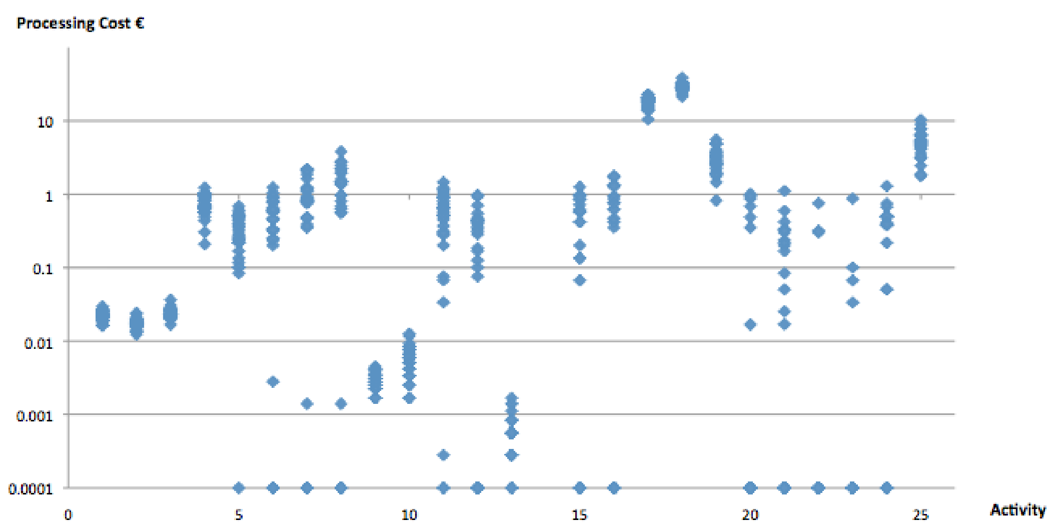

Figure 7.11: Processing Cost 25 Users

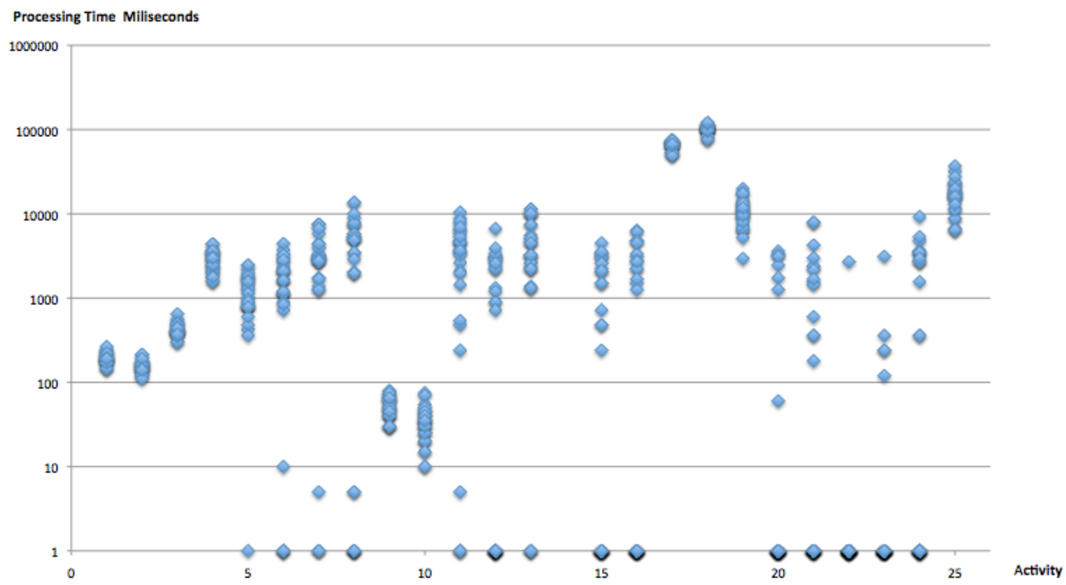

Figure 7.12: Processing Time 25 Users

The data from the above charts was generated by simulating 30 times the usage of 25 users of a virtual STB in a cloud environment for the period of 24 hours. We have ran every simulation 30 times and aggregated the results in a single file so it would have statistical relevance.

We didn't think that our work would be done if we just simulated 25 users in a cloud environment, because that would be a very small sample to test our implementation. So we decided to see how would the simulation environment behave and how would the cost and processing time be like if we increased the number of users. We incremented the number of users by 25 in each run until we got to 100 users.

We have plotted the charts for cost and processing time for each of the increments(25, 50, 75 and 100 users). We believe that it would be very monotonous for the readers of this work to look a very similar charts so we chose to show the data regarding the initial simulation of 25 users and then make some comparisons regarding the activity cost and user cost for each increment.

7.7.3 User Cost

If we take a look at Figure 7.13 we can see how an increase in the number of users has led to the optimization of resources. The cost has decreased significantly with the increase of users. Comparing the cost decrease from 25 to 100 users the value has changed from €2.45 to €2.3.

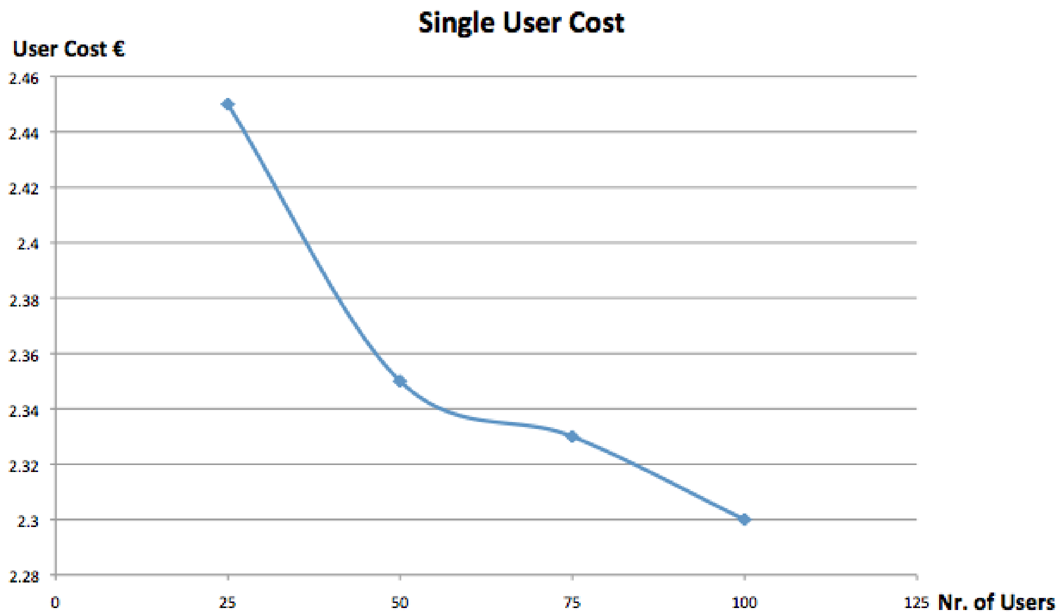

Figure 7.13: User Cost Evolution

Bear in mind that this price is for a single user, in the universe of 100 users, interacting with a virtual STB in the cloud for the period of one day. We believe that with the economics of scale we could decrease the price even further. By taking advantage of special discounts CC providers would give us and the optimization of resources due to the elevated number of users we believe this would be possible and economically feasible.

7.7.4 Active users and sessions

After having reached the previous results we wanted to confirm if the data that we generated with our log Synthesizer could be compared to the result that we have analyzed previously [9, 10].

We wanted to view if the number of active users would follow the same pattern we found while doing the meta-analysis of [9] and [10].

Taking a look at Figure 7.14 we can confirm that the generated data followed the pattern of growth throughout the course of a day and started declining at the end of the day.

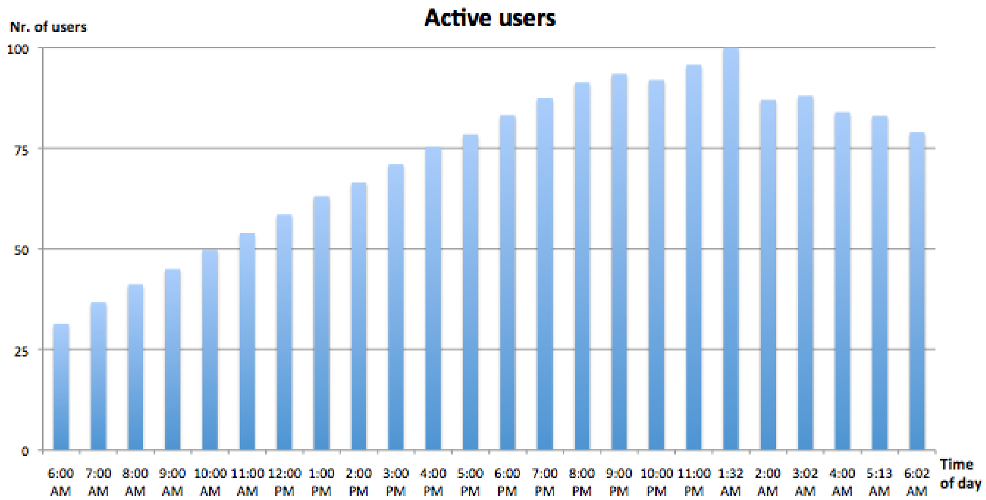

Figure 7.14: Active Users

The utilization of the IPTV during the morning is quite low, it starts increasing and does so throughout the day up until it reaches it's peak and then starts decreasing again. This is quite similar to the pattern we have discovered during the meta-analysis of [9] and [10].

We also calculated the Pearson coefficient to get even further confirmation of what we have stated previously. The result we got was 0.95 when we compared our results to [10].

We were quite happy with these results despite not being exactly the same as the ones in [9] and [10], they allowed us to prove what we set to accomplish, that it is possible to simulate the behavior of users without having access to the logs from the ISPs.

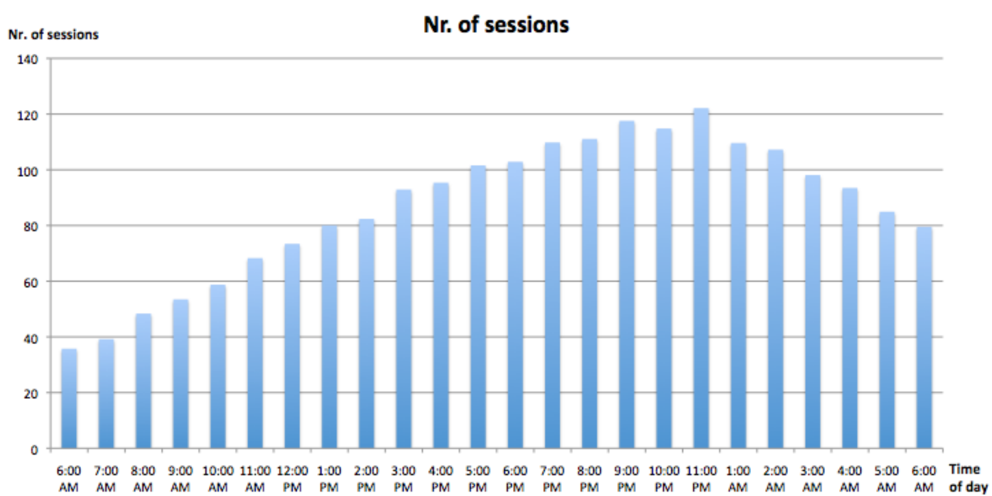

Figure 7.15: Number of Sessions

After having reached the results we desired we wanted to verify if they were actually correct, so we decided to map the number of active sessions to make sure that we where having consistent results. As seen in Figure 7.15, the number of active sessions also follows the same pattern of growth throughout the day and decreases after having reached its peak.

We have found a small bug during the mapping of the previous charts, there is a gap between 11:00 PM and 01:00 AM, probably it is a bug on our log analyzer that we have to solve in the future.

7.7.5 Activity Cost

The same pattern of cost decrease applies to the activities that are more expensive to process by the simulator. While looking at the total cost of the VOD Figure 7.16 and DVR Figure 7.17 activities we can see that by increasing the number of users we also decrease the total activity cost.

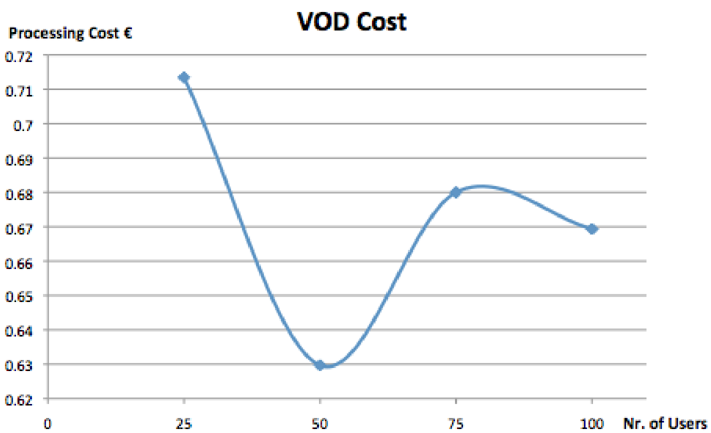

Figure 7.16: VOD Activity Cost

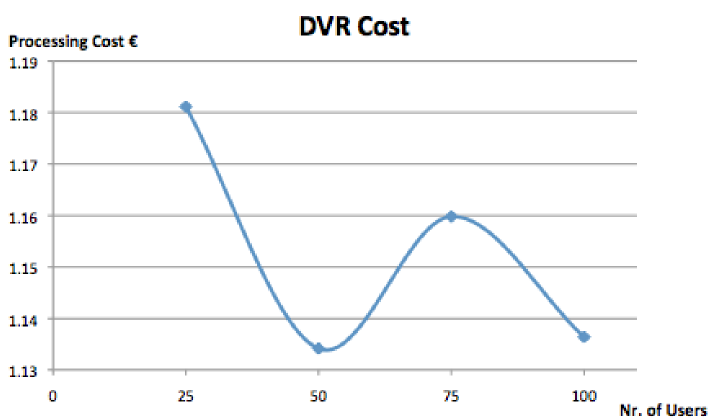

Figure 7.17: DVR Activity Cost

There's a slight fluctuation when the number of users increases from 50 to 75 users, but that changes when we increase the number of users to 100.

Once again, by demonstrating the cost evolution of the most expensive activities, we have proved that the optimization of the equipment takes place when we increase the number os users.

Both the total user cost and de most expensive activities cost decrease by increasing the number of users. This leads us to the conclusion that there is an optimization of resources and that it is possible to accomplish a economically feasible implementation of our STB in the cloud.

7.7.6 Energy Usage

The usage of energy increases with the amount of users that make use of the cloud environment. This is logical because with the increasing need of processing power the datacenter will also increase it's need of energy for the utilization of CPU (Figure 7.18).

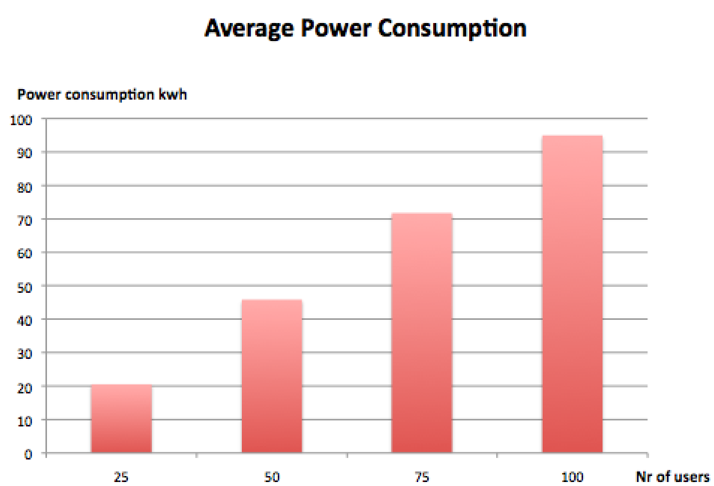

Figure 7.18: Average Power Consumption

As we can see in the previous charts the average energy consumption grows linearly to the amount of users that are connected. The power consumption floated during our simulations, in the following chart (Figure 7.19) we can see the highest and lowest amount of spent energy while simulating our environment.

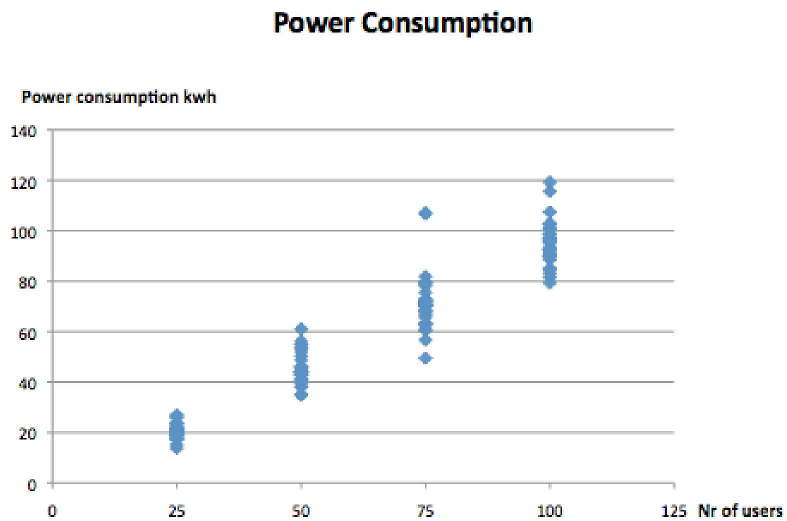

Figure 7.19: Power Consumption

The power consumption increases if we increase the number of users. For 25 users we can see that we have an average power consumption of 20kwh, for 50 users we have 45kwh, for 75 users 71kwh and finally for 100 users we get 101kwh this leads to the conclusion that for each user the system needs approximately 0.95kwh.

7.8 Summary

In this chapter we have achieved the objectives we have set in the first place. We were able to use the platform we chose and managed to use it's several classes on our behalf. We overcame our difficulty to obtain user logs from behalf of ISPs and created our own log synthesizer. We also were able to generate a simulation environment that allowed us to extract some data regarding the utilization of a CC environment for the deployment of a virtual STB. Data like the energy consumption, processing time and processing cost of activities proved helpful in understanding how could our system behave in a real life situation. After having analyzed the data regarding the cost and energy consumption we focussed on the number of active users and sessions across the simulation. The study of these two last pieces of information allowed us to prove that our Log Synthesizer generated realistic looking data.

Chapter 8

Conclusions and Further Work

In this chapter we want to discuss the results we have obtained and give some thoughts about our work. We will start by discussing the results we have reach during our simulations.Then we will make an analysis to our work, describing our achievements,some setbacks and the possibilities for future work.

Finally we will discuss our conclusions.

8.1 Obtained results

Our desire was to prove that it would be possible and economically feasible to move the STB to the cloud. For us to be able to discover if it was cheaper to run the STB in the cloud we had to simulate the behavior of the equipment in a cloud environment.

We have created a STB model and implemented it using cloudlets so we could simulate it.

We used CloudSim to simulate the platform and created cloudlets similar to the activities a STB has to preform. Those cloudlets would require the same amount of processing power and time as they would do in a real STB.

The creation of a Log Synthesizer based on previous works we have found [9, 10] was the only solution to the fact that we did not have access to real logs from ISPs.

We decided to model each STB as a virtual machine in the simulation environment so we could obtain some results that could help us in our work.

We were able to prove that the cost per user lowers if we increase the number of clients. We also identified the most expensive activities during the course of the simulation.

We simulated a cloud environment with 25, 50, 75 and 100 users(every increment was simulated 30 times) and got some results regarding the cost of processing, the time it took to process and the energy consumption.

By making some calculations, as explained in the previous chapter, we have come to the amount of €27.10 fixed monthly cost per VM.

The variable cost was calculated by using data from the simulation were we had 100 users, since this is the one we had better results with.

The average daily cost is of about €2.3, but we believe that a better value can be reached if we make use of the economies of scale during the procurement of CC providers. This can be seen in the previous chapter where we can distinctly see a decrease in the cost per user.

In this first approach to this problem we have defined that each VM is a single STB, so every user is associated to a single VM when he is connected to the service. We believe that we can improve on this aspect of our work because we can optimize the connections and activities users are executing at the same time.

In our current implementation if two or more users are viewing the same channel they are connected to different virtual machines that are getting the same IPTV stream. This also happens when several users are recording the same TV show or viewing the same VOD. In the next iteration of our work we would like to optimize the resources even further by redirecting requests that are similar to a single virtual machine or a group of virtual machines that can be dedicated to specific tasks. By implementing such a system we could remove the need for a VM for each user.

The monthly cost of using the CC platform for a whole month would be substantial if we are talking about only 100 users. Since it is impractical for an ISP to only have such a small amount of costumers, we believe that increasing the amount of users to a couple of thousands and even a few millions in our simulation would lead to better results.

The computational requirements for the execution of simulations with such a high number of users far exceeds the capability of the equipment we had available. That is why we have reasoned to stick to the results we have reached during the development of our work.

In our simulaton, the total amount a user would pay each month would be approximately €96.10 (daily variable cost of €2.3 plus the fixed cost of €27.10).

We believe that the results we have achieved are expensive, compared to the value ISPs practice in Portugal where for about €45 a month you can get 100 TV Channels, Internet service and landline telephone. The results we have reached cost almost double the price some ISPs are practicing, and only for the usage of the cloud to store the STB.

The final value that we have reached might have been influenced by averaging the prices of storage and bandwidth cost, if we had chosen the cheapest value for each one of them we might have gotten different results.

We believe that in a real environment we could reach a lower price if we took advantage of the economies of scale. In our simulations we have used a sample of 100 users, if we were to transport this to a real life situation we could have thousands and even millions of users paying for our services. With such a high number of users we would be able to have some leverage to negotiate better prices with Cloud Computing providers for the usage of their infrastructure.

8.2 Discussion

We believe that our work is better discussed and comprehended if we make an analysis of what we have accomplished. This analysis will help us to see what we did, what we missed and what we can improve. At the same time this can lead to the discovery of developments to be made in the future.

One of the major achievements we can identify in our work is the utilization of a simulation platform that is not very well documented. We have dedicated some time to study it and show in a very simple way, how can we use it to create a simulation, what are the major classes and how they interact.

Another achievement that we believe is worth being mentioned is the creation of our own log synthesizer. With it we were able to generate logs that are very similar to the ones the ISPs have available. By having created this tool we were able to be more independent in the course of our work. Because we did not have any sponsorship or guidance from any ISP and we were able us to pursue our work the way we saw more fit.

We believe that our proposal for the removal of the STB from the living room and move it to the cloud is an achievement in our work. Even if we did not prove that is more economical to have the STB in the cloud, we were able to raise some awareness to the possibility of it occurring and demonstrate that by increasing the number os users we could get better results. This is the example of an another achievement we were able to turn into a possibility for future work.

We have talked about our achievements but believe we also should talk about the setbacks and limitations we have encountered.

We believe that time was one of our main setbacks. Time to work on our project, time to organize the data we had and time to produce this document.

The utilization of the simulation environment is an achievement and a setback at the same time. A setback because it took us a lot of time to discover how it worked and how to change it to accomplish what we wanted. And an achievement because we were able to discover a way to work with it and show others how it can be done.

The lack of real logs to help us with the simulation of our environment was another limitation that we were able to turn into an achievement. We tried to get logs from several ISPs and even from authors that had worked in similar subjects, but we had no luck, so we decided to create our log synthesizer.

We have talked about our achievements , setbacks and our limitations, so let us talk about the menaces to the conclusion and improvement of our work.

Once again, time. Since ”time is of the essence”, meeting our deadline was one of the menaces we faced. Whether it was because of the lack of time due to personal or work related problems.

We also believe that the timing for the conclusion of this work has lead us to be a little far off what we thought we would be. When we started our work no one had heard of the possibility of using the cloud for IPTV services, but now some ISPs are already offering that possibility and that's why we think timing is a menace to our work.

The results we have reached are also a menace that we have found. This is so because the cost of moving the STB to the cloud is not as cheap as we thought it would be. Those results could have been better if we had a real test environment and took advantage of economies of scale but that was already discussed.

Our final observations go to the possibilities for improvement and future work that we believe can be achieved.

One of the major opportunities we see in our work is the possibility to improve it. Wether it is by developing a a new log synthesizer or by rewriting our simulation code. We believe that we have space to grow this work and that our findings are just the tip of the iceberg.

The creation of our log synthesizer is another opportunity for further work we have discovered, since there is no literature regarding this subject we believe that this can make our work helpful to other people investigating the behavior of users and STBs in a IPTV and CC environment.

In this work, we made a simplistic approach and we have a correspondence between each set-top box and a virtual machine. So if we have N STBs, we have N virtual machines (VMs).

However, if one is working with thousands or even millions of STBs, there is an high probability to have large subsets performing the same activity.

In such case, instead of establishing a 1:1 relationship between set-top boxes and virtual machines, it is more productive N:1 relations between STBs and a virtual machine which performs the respective activity. Each time a STB starts an activity is added to the VM list. If it ends the activity is removed from the list. Obviously, there is an activity record per set-top box.

If there are many users to do the same activity, one may need to have M virtual machines where the activity takes place. Thus, we will have greater scalability because for each N set-top boxes we will have K virtual machines, with K $\ll N$.

The possibility of changing our platform to perform as stated is also an opportunity for future development of our work.

Finally, our utilization of CloudSim and the auxiliary tools we have made so it could simulate a STB in a CC environment is also an opportunity for further work. We can use our existing tools, feed them some real logs and simulate their behavior in a CC environment.

8.3 Final Remarks

CC is not "the silver bullet" of high performance computing but it is on it's way of getting there.

We believe that the lack of standards, some lack of security and fear of vendor lock-in are the major reasons why it is not growing as it should be.

IPTV has come a long way since the beginning of this work and is capable of providing interaction with the users as we have never seen before, but it is missing that extra step towards being a service that can be accessed everywhere.

We believe that by joining these two technologies there's a possibility to create a major breakthrough on the entertainment market. The possibility to view your favorite TV shows, recorded series and movies anytime and anywhere. View a video or other kind of program at a friend's or relative's house without the need to be on the same network provider. These are some of the possibilities that the junction of these two services could provide us.

Our work focused on simulating the utilization of a virtual STB in a CC environment and retrieving the results regarding the costs of processing all the activities generated by the users interaction with the equipment through the network. After having analyzed the result we discovered that our initial idea of reducing the costs was not very correct since the value we have come up with are more expensive then the ones being currently practiced by ISPs.

Despite these results we believe that they are not very far off, they are of the same order of magnitude as the prices that are currently being practiced by ISPs. We believe that they can decrease even further if we take advantage of the economies of scale.

We believe that our work can be further improved by reviewing the code that maps the users activities to the cloudlets that are processed by CloudSim and by adding some extra features to our log synthesizer so it would generate activities in a more realistic way.

Bibliography

[1] M. Coward, "Addressing the bandwidth demands of IPTV." Web, January 2010. http://www.ccpu.com/news/articles/200709-iptvbandwidth.html.

[2] G. Bertrand, "The IP Multimedia Subsystem in Next Generation Networks," *Rapport technique, ENST Bretagne*, 2007.

[3] E. Mikoczy, D. Sivchenko, B. Xu, and J. Moreno, "IPTV Systems, Standards and Architectures: Part II - IPTV Services over IMS: Architecture and Standardization," *IEEE_M_COM*, vol. 46, pp. 128–135, May 2008.

[4] *IPTV Focus GroupProceedings*, ITU-T, 2008. http://www.itu.int/publ/T-PROC-IPTVFG-2008/en.

[5] F. Silva, N. Badrudino, and T. Sousa, "Iptv–arquitecturas, protocolos e serviços." Instituto Superior Técnico-Taguspark–Portugal, dissertação de mestrado, 2007.

[6] S. Vanhastel and R. Hernandez, "Enabling iptv: what's needed in the access network," *Communications Magazine, IEEE*, vol. 46, no. 8, pp. 90–95, 2008.

[7] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an ip network," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pp. 71–84, ACM, 2008.

[8] G. Yu, T. Westholm, M. Kihl, I. Sedano, A. Aurelius, C. Lagerstedt, and P. Odling, "Analysis and characterization of iptv user behavior," in *Broadband Multimedia Systems and Broadcasting, 2009. BMSB'09. IEEE International Symposium on*, pp. 1–6, IEEE, 2009.

[9] C. Silva, "USER BEHAVIOR IMPACT ON IPTV PLATFORM PERFORMANCE," Master's thesis, Faculdade de Ciencias da Universidade de Lisboa - Departamento de Informatica, 2010.

[10] T. Qiu, "Understanding a large-scale iptv network via system logs," Master's thesis, Georgia Institute of Technology, 2011.

[11] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," *Arxiv preprint arXiv:0901.0131*, pp. 1–10, 2008.

[12] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[13] R. Buyya, R. Ranjan, and R. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*, pp. 1–11, IEEE, 2009.

[14] A. Odedra, "ITU-T NGN Standardisation," 2006. 8 March 2006 Paris OECD ICCP Workshop. Future of Internet.

[15] "The cochrane collaboration open learning material - module 12: Combining studies." The Cochrane Collaboration 2002.

[16] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, *et al.*, "Sip: session initiation protocol - rfc 3261," tech. rep., RFC 3261, Internet Engineering Task Force, 2002.

[17] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter base protocol - rfc 3588," tech. rep., RFC 3588, September, 2003.

[18] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The cops (common open policy service) protocol - rfc 2748," tech. rep., The Internet Society, 2000.

[19] S. E. Deering, "Host extensions for ip multicasting," *ietf.org*, 1988.

[20] B. Fenner, H. He, B. Haberman, and H. Sandick, "Internet group management protocol (igmp)/multicast listener discovery (mld)-based multicast forwarding ("igmp/mld proxying")," 2006.

[21] M. Volk, J. Guna, A. Kos, and J. Bester, "IPTV Systems, Standards and Architectures: Part II - Quality-Assured Provisioning of IPTV Services within the NGN Environment," *Communications Magazine, IEEE*, vol. 46, pp. 118–126, May 2008.

[22] J. Asghar, F. Le Faucheur, and I. Hood, "Preserving Video Quality in IPTV Networks," *Broadcasting, IEEE Transactions on Volume 55, Issue 2, June 2009 Page(s):386 - 395*, vol. 55, pp. 386–395, June 2009.

[23] M. Zapater and G. Bressan, "A proposed approach for quality of experience assurance of iptv," in *Digital Society, 2007. ICDS'07. First International Conference on the*, pp. 25–25, IEEE, 2007.

[24] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1103–1120, 2007.

[25] T. Wiegand, L. Noblet, and F. Rovati, "Scalable video coding for iptv services," *Broadcasting, IEEE Transactions on*, vol. 55, no. 2, pp. 527–538, 2009.

[26] C. Lee, C. Hong, and K. Lee, "Reducing channel zapping time in iptv based on user's channel selection behaviors," *Broadcasting, IEEE Transactions on*, vol. 56, no. 3, pp. 321–330, 2010.

[27] F. Ramos, J. Crowcroft, R. Gibbens, P. Rodriguez, and I. White, "Channel smurfing: Minimising channel switching delay in iptv distribution networks," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pp. 1327–1332, IEEE, 2010.

[28] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," *NIST special publication*, vol. 800, p. 145, 2011.

[29] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[30] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

[31] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, "The characteristics of cloud computing," in *Parallel Processing Workshops (ICPPW), 2010 39th International Conference on*, pp. 275–279, IEEE, 2010.

[32] I. Sriram and A. Khajeh-Hosseini, "Research agenda in cloud technologies," *Arxiv preprint arXiv:1001.3259*, 2010.

[33] N. Borenstein and J. Blake, "Cloud computing standards: Where's the beef?," *Internet Computing, IEEE*, vol. 15, no. 3, pp. 74–78, 2011.

[34] J. Myers, "Local mail transfer protocol - rfc 2033," *ietf.org*, 1996.

[35] N. Freed and N. Borenstein, "Multipurpose internet mail extensions (mime) part one: Format of internet message bodies - rfc 2045," 1996.

[36] M. Butler, J. Postel, J. Reynolds, D. Chase, and J. Goldberger, "Post office protocol: Version 2 - rfc 218," *ietf.org*, 1985.

[37] P. W. Resnick, "Internet message access protocol (imap) catenate extension - rfc 4469," *ietf.org*, 2006.

[38] A. Zahariev, "Google app engine," *Helsinki University of Technology*, 2009.

[39] https://developers.google.com/appengine/.

[40] https://code.google.com/p/appscale/.

[41] https://aws.amazon.com/ec2/.

[42] http://www.eucalyptus.com/.

[43] http://www.windowsazure.com/en us/.

[44] http://www.ibm.com/cloud computing/us/en/.

[45] https://www.salesforce.com/eu/?ir=1.

[46] K. Breitman, M. Endler, R. Pereira, and M. Azambuja, "When tv dies, will it go to the cloud?," *Computer*, vol. 43, no. 4, pp. 81–83, 2010.

[47] E. Abd-Elrahman and H. Afifi, "Moving to the cloud: New vision towards collaborative delivery for open-iptv," in *ICN 2011, The Tenth International Conference on Networks*, pp. 353–358, 2011.

[48] Y. Wu, C. Wu, B. Li, X. Qiu, and F. Lau, "Cloudmedia: When cloud on demand meets video on demand," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pp. 268–277, IEEE, 2011.

[49] S. Kanrar, "Analysis and implementation of the large scale video-on-demand system," *Arxiv preprint arXiv:1202.5094*, 2012.

[50] R. Lavanya and V. Ramachandran, "Cloud based video on demand model with performance enhancement," *Malaysian Journal of Computer Science*, vol. 24, no. 2, p. 73, 2011.

[51] V. Aggarwal, X. Chen, V. Gopalakrishnan, R. Jana, K. Ramakrishnan, and V. Vaishampayan, "Exploiting virtualization for delivering cloud-based iptv services," *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference*, pp. 637–641, 2011.

[52] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. Khan, “Greencloud: a packet-level simulator of energy-aware cloud computing data centers,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, IEEE, 2010.

[53] R. Buyya and M. Murshed, “Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing,” *Concurrency and Computation: Practice and Experience*, vol. 14, no. 13-15, pp. 1175–1220, 2002.

[54] T. Rodrigues, “Side-by-side comparisons of iaas service providers.” Web, August 2012. http://www.techrepublic.com/blog/datacenter/side-by-side-comparisons-of-iaas-service-providers/5717.