Pre-fetching and Caching in Catch-Up TV Network

WAEL JADDOA
HOUSAM ABBAS
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Master's Thesis

Pre-fetching and Caching in Catch-Up TV Network

by

Wael Jaddoa (Wireless Communication)

Housam Abbas (Computer Science)

Supervisors:

Jens Andersson Stefan Höst

Examiner:

Maria Kihl

LUND UNIVERSITY

Department of Electrical and Information Technology

Faculty of Engineering LTH

Lund University

2017

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Nowadays, huge amounts of data are transferred over the Internet. Video content is a vast majority of the data traffic. A part of this data is wasted due to many reasons. Transmitting large amounts of data across a network leads to network congestion and start-up delay of the video playback, which is one of the reasons of data loss. Another reason is the negative effect of impatient users or so-called zappers. This effect is expressed by the unexpected change in user preference during the streaming process of the requested content. In that case, the downloaded data are discarded which can be considered as a waste of network and system resources.

One of the solutions to this problem is to develop a system for prefetching and caching, which is used to prepare the expected contents to be ready for any possible request by the users. The improvement in the system is approached by eliminating excess data transfer, and by caching needed parts of the contents. Thus reducing the load on the network and saving the resources.

The aim of this study is to investigate user behavior and define user groups as Loyals or Zappers on a scale of a grading system. This is performed by analyzing collected data of user requests and content details to find a way to adjust prefetching and caching system settings based on several factors; user behavior, session length, content length, and content popularity. An individual calculation for each of these factors is done to get the specified results which are shown in graphs to have an overview of the analyzed data, and to extract useful information to reach the goal of this work. All of the studied subjects are contributed to produce an enhanced model of the prefetching and caching system.

After demonstrating the results, some variable values can be evaluated from the calculations. These values vary depending on the processed data. That would affect the accuracy of the outcomes.

Popular Science Summary

Internet TV has been more available and widely popular recently. A fast Internet broadband connection, which can deliver high-quality videos, is one of the reasons for the flourishing of the Internet Protocol TV and Video-on-Demand service. Another reason is the possibility to watch TV contents from around the world over the Internet, at any place in the world and at any time. Catch-up TV is a service which permits the viewers to access TV programs and video contents beyond the scheduled original broadcasting time.

Prefetching and caching techniques have been used to reduce the start-up delay and the latency in viewing online video streams. Prefetching is predicting the possible videos that the user may be interested to watch next, and preload a part of them in the cache, before they have been requested by the user. The cache is a fast memory located at the server, and also can be located at the end-user's terminal. A part of the frequently requested video, or the predicted video, can be stored in the cache. The benefit from video caching is to decrease the delay of retrieving the videos from the main storage of the server, or from its original source, and to make the videos ready to be viewed by the user in a fast way.

The first issue that faces the Catch-up TV service is the unused data transfer. Preloading a lot of videos that may not be watched by the user, is a waste of the system resources. As a result, the bandwidth will be exhausted, and the cache will be filled with unnecessary data. The second issue is the increasing usage of the Internet TV service that can cause server overload with many requests from the users, and leads to network congestion, data loss, and degradation of video streaming quality.

Analyzing the usage of the service has an emerging importance to enhance the service by allocating system and network resources based on the behavior of the users and the requirements of the service provider. The users do not have the same interest in the available video contents of the service, many users start watching a video, then leave it right in the beginning, and change to another video, other users continue watching the requested video to the end. The impatient users, who constantly change their watching preference at the beginning of the video stream, have a great impact on the network, that is a lot of data will be transmitted over the network with no use. The unused data forms an excessive load on the server and the network. The varied user activity should be taken into the account when deciding system settings, and when distributing system and network resources.

The aim of this project is to analyze user behavior in catch-up TV network in order to identify user viewing habits to provide the required resources to each user. We strive to

find an adaptive caching and prefetching algorithm that can be implemented to decrease the amount of cached data in the server and the user cache, and also to decrease the transmitted data over the network. Studying user behavior and trying to use new methods of prefetching and caching techniques, achieves twofold benefits, both for the service provider by saving system and network resources, and also for the users by improving video streaming quality and decreasing time delay.

There are many factors that can be used to achieve the goal of this research. Those factors are related to both user behavior and content properties. The factors related to the user concern the preference of the user based on the viewing history of the user, and also about how long the user watches the contents in average. On the other hand, the factors related to the contents concern its popularity, length, and type.

We have studied each factor separately to one value denotes the result from each studied section that summarizes the user activity or the content properties. The resulted value from each section can be combined together to get an ultimate result to decide how much of the data is needed to be cached for each predicted content, and how much data is needed to be transferred to each user. All those factors can be used cooperatively to determine the right amount of the data to be cached in the server cache and in the user cache. The cache of the server is more related to the common contents, which are viewed by many users. Furthermore, the cache of the user is related to the activity of each specified user.

Since different users have different preferences, which can change all the time, even for the same user, the data analysis should be done in real time in the server to achieve an actual outcome that can be used to modify the setting of the system to get a better result.

Preface

This master thesis is written by Wael Jaddoa and Housam Abbas, with support from our examiner Maria Kihl, and our supervisors, Jens Andersson and Stefan Höst, Electrical Information Technology (EIT), Lund University. We both worked together on the entire tasks of this work. Our tasks have begun in gathering the needed information about the scientific background and related works. After that, we have collected the needed data and analyzed it to get proper solutions to the researched problems. The analysis is done by writing code in Python (programming language) to import the data, analyze it and obtain outcomes. The obtained results are presented and discussed in this report, and also plotted in many charts. At the last, we wrote this thesis, which introduces our work. We hope it comes up to the reader's expectations.

Table of Contents

List of Figures

List of Tables

Glossary

ARC Adaptive Replacement Cache, a caching algorithm which considers both frequency and time of use when updating cache objects. 10, 25

Buffering Collecting data in a temporary memory and preparing it for transfer over the network with adaptive speed. 1, 3, 9, 25, 29, 30

Cache Fast memory or disk buffer intended to store data for future use. 1, 3, 4, 10, 11, 14–17, 20, 22, 23, 25–30, 32–37, 40, 42, 45–48

Caching Passively saving the requested content in a cache to be used for reducing the latency when the content is requested again. 1–3, 9–11, 13, 14, 16, 17, 25, 29, 35, 36, 42, 45–47

Catch-up TV Viewing already started programs or previously aired episodes of a TV show at any time. 3, 8, 17

CDF Cumulative Distribution Function, a continuous distribution forms the probability up to a specific value. 29, 33, 37, 40, 43, 44

CPV Content Popularity Value is a single value represents the popularity of the content. 16, 20, 26, 27, 32, 33

CSV Content Session Value, represents the amount of seen part of the content. 16, 22, 23, 26, 27, 45

HTTP Hypertext Transfer Protocol is an application protocol used to provide information on the Internet. 6

IPTV Internet Protocol TeleVision is delivering TV broadcast over the Internet using dedicated servers and user appliances. 5, 6, 10, 14, 21, 23, 24, 27, 29

LFU Least Frequently Used, a caching algorithm which discards the content with the least usage time. 10

Loyal A user who keeps watching TV programs to the last minute, or most of its length. 1, 2, 11, 15, 16, 21, 23, 24, 27, 36, 41–43, 46

LRU Least Recently Used, a caching algorithm which discards the content with the oldest used contents. 10

Matlab Matlab is a numerical computing software and programming language. 13

NaN Not-a-Number, an empty value corresponding to something with no value or null in programing languages. 14

NPC Next Possible Content, is the expected contents to be viewed by the user later. 15

Numpy Numeric Python is an additional package for the Python programming language that deals with mathematical functions. 13

OTT Over the Top, is a technique of delivering multimedia over the Internet without using special or proprietary devices. 5, 6

P2P Peer-to-Peer is a data sharing method that involves two or more users that cooperatively shares portions of data directly from one user to another one without the need for a central server. 9

P2PTV Peer-to-Peer TeleVision, Watching TV stream using P2P network and sharing the stream with other users at the same time. 9

Pandas An open source library which works within Python programing language, and provides fast and efficient data analysis tools. 13, 14

PF Prefetching Factor, a value that decides which contents should be cached in a specific user cache based on the viewing history of that user. 26, 27

Prefetching Actively predicting and pre-loading the next expected content based on the previously seen contents to reduce the buffering delay from the first use. 1–3, 9–11, 13–17, 24–29, 42, 45–47

Python Python is an object oriented and structured programming language. 13, 14, 17, 29, 39

QoE Quality of Experience, scale of network performance and reliability of network characteristics measured by the provider. 1, 5, 6, 11, 14, 15, 23, 29, 45, 47

QoS Quality of Service, scale of expected network quality and service performance from the user point of view. 1, 3, 5, 6, 14, 15, 23, 29, 45

RTP Real-time Transport Protocol, a protocol used in streaming video and audio data over the Internet. 6

SCS Server Cache Setting, decides how server cache is managed and regulated based on contents popularity and average session length. 26, 27

Spyder An open source software for scientific programming in the Python language. 13

SQL Structured Query Language, a programming language used to manage and control data in the database. 14

SVOD Subscription Video-on-Demand is a TV/film paid service where the user has to subscribe to this streaming service to be able to get access to a large amount of video content. 7, 8

TCP Transmission Control Protocol is a transport protocol using to transmit the data. 6

UBV User Behavior Value denotes a single value indicates user watching time. 15, 23–27, 41–43, 45, 46

UCS User Cache Setting, a value that decides how the user cache is managed and regulated based on user session length and user viewing history. 26, 27

UDP User Datagram Protocol is a transport protocol used to transmit the data over the Internet. 6

VOD Video-on-Demand is a service used to stream multimedia contents over the Internet at any time when requested by the user. 5, 7, 8, 10, 11, 27, 37

VoIP Voice over Internet Protocol is a service used to communicate with voice over the Internet as an alternative to the regular telephone network. 6

Zapper An impatient user who switches between programs and channels often, and does not have a fixed preference. 1, 2, 5, 10, 11, 14–16, 21, 23, 24, 27, 35, 42, 43, 46

Chapter 1
Introduction

The current Internet infrastructure might not be capable of taking massive requests for Internet-based TV and video broadcasting, especially when talking about video loading latency and network congestion. Therefore, new studies are needed to enhance QoS of the network and QoE for users. Prefetching and caching are some of the methods that could be used to reduce the start-up delay which is the time measured from the request for a content until it is ready to play at the end terminal. Prefetching is actively predicting and pre-loading the next expected content based on the previously seen contents. Caching is passively saving the requested content in a cache to be used for reducing the latency when the content is requested again. On the other hand, prefetching would effectively reduce the buffering delay already from the first request.

1.1 Objectives

The objectives of this project can be summarized as follows,

- The first goal is to define user groups according to prefetching and caching strategies of adaptation to each user profile. Two user groups can be defined, the first one is Loyal users, and the second one is impatient users (Zappers). However, there are some users who may sometimes change their behavior from one group to another.
- The second goal is to study each group behavior, finding a good method to simulate and model these groups and represent their usage activity anonymously with statistical graphs that help to understand user behavior.
- The third goal is to represent the effect of Zappers on prefetching and caching, and to try to find a good solution to this problem by reducing their impact and speeding up the prefetching process efficiently.

1.2 Problem Formulation

In this study, some research questions need to be answered. The first issue is to define user groups according to their behavior, and identify users of the Internet based TV service as Zappers or Loyals. The second issue is to identify the impact of Zappers on network traffic, and find a good solution for this issue. The main problems that should be handled are network congestion and time delay, which lead to a low quality video streaming. One thing that needs to be considered is to design a prefetching system, and implement the best caching algorithm that can be used to reduce data traffic and latency. There are several reasons why it is good for science to do this project.

- The first reason is improving network performance and avoiding network congestion by eliminating unnecessary data transfer, which would help keep the network stable.
- The second reason is reducing start-up delay of the requested content by implementing a good prefetching strategy and decreasing latency by caching the correct content.
- The third reason is to obtain a clear image of user behavior for future improvement to have a better method that handles random user browsing activity or so-called Zapping.

1.3 Thesis Outline

The outline of the thesis work is presented in this section. The thesis report consists of seven chapters. A brief description about each chapter is written below.

- Chapter 1 Introduction: presents project objectives, problem formulation, thesis outline, limitation in the data set, and system structure.
- Chapter 2 Background: reviews background of the thesis, related work, and definition of concepts.
- Chapter 3 Methodology: describes the used methods, software and interface. It also includes a description about the prefetching system.
- Chapter 4 Data Analysis: contains the analysis of the data set and the equations used in the calculations.
- Chapter 5 Result and Discussion: discusses the terms and shows the outcome figures and their connection to problem formulation.
- Chapter 6 Conclusion: concludes the report with a short summary.
- Chapter 7 Future Work: contains some suggestions for future work.

1.4 Limitation

There is a limitation in the data set that prevents us from knowing the zapping time, which is the period of time between the start time of the user request and the start time of the video playback. Another unmeasured parameter is the buffering time, which is the time between receiving the first packet of a video stream and starting viewing the video by the user. The given data set has only a timestamp of the user request but there is no timestamp of the first received packet of video stream nor the video viewing start. Therefore, network delay cannot be calculated from the given data set and also the buffering time is not available. This limitation makes it harder to directly know if the suggested strategies of caching and prefetching will actually make a large enhancement in the performance of Catch-up TV network and service, or only a slight boost in reducing delay time.

1.5 Caching System Structure

The main purpose of using the cache is to reduce the latency, and to maintain streaming performance as well as to increase QoS of the network. Another benefit of the caching system is to reduce data traffic, since it can provide previously requested data directly to the users without the need for a new download from the server over the network.

Figure 1.1 shows a simple scheme of the caching system structure. This structure shows the path of the data traffic and displays basic network connections. It also shows that data caches can be placed on the server side and the user side. It is supposed that the original data contents are imported from the data storage or from a remote server.

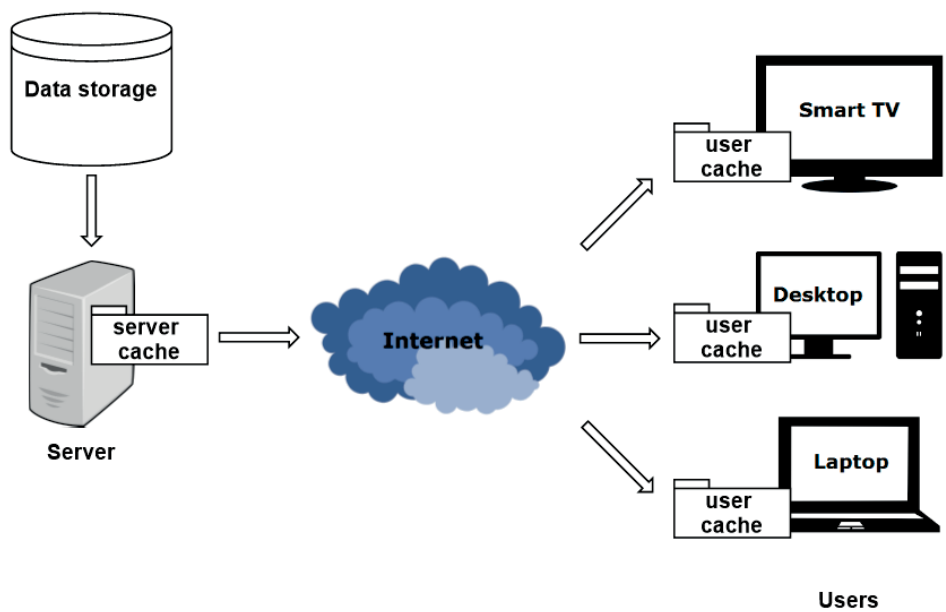

Figure 1.1: Caching system structure

The server cache can be used to store some important data contents ready to be served to the user over the Internet or a local network. This will reduce the time delay encountered when requesting the content from its original storage place. Furthermore, a proxy server or several local servers can be added to the system to have extra support to the system with more cache memory. This will help to move the needed data near towards the users in order to reduce the time delay in reaching the requested data.

User caches are used to store the expected contents which are going to be watched by each user in its own local cache. This will introduce a reduced latency of video streaming. At the same time it can increase the data traffic on the network if it is not performed in a correct way. The best solution is to cache just the needed parts of the contents at the right time when the there is low load on the network.

Chapter 2
Background

Multimedia streaming has become an important and widespread feature in the Internet. With the increasing demand for this service, more challenges appear in delivering high quality video and audio content to the users over the current network infrastructure. Improvement in network construction and protocols is required to increase QoS and QoE.

Zapping is one of these problems that can cause network congestion and limit the available bandwidth. Zappers are a kind of users who change the channel often within a short period of time as they tend not to complete the whole requested video stream. Zapping time is another parameter that can be defined as the waiting time delay until the requested video stream is ready to start playing. It will be unpleasant to the user if the zapping time is too long [1].

2.1 Definition of IPTV, OTT, and VOD

Internet Protocol Television (IPTV), Over-the-top (OTT), and Video-on-Demand (VOD) are services used to stream multimedia contents over the Internet. In order to understand and discuss any research problems in the field of Internet TV we have to define each service and mention the differences between these services.

IPTV is delivering TV broadcast over the Internet using dedicated servers and user appliances, but OTT is delivering multimedia over the Internet without using special or proprietary devices. VOD users can select a video from a list of available videos, and watch it at any time. All of these services are transmitting video and audio data carried by network packets using the Internet Protocol.

IPTV, OTT, and VOD are introducing modern and popular methods to view TV channels and multimedia over the Internet instead of traditional analog or digital terrestrial, cable or satellite TV broadcasting. However, these new methods are also delivered via the same medium in a more efficient way, using data packets routed through broadband connection networks over the World Wide Web in an interactive way. The user can choose and request a desired content to be viewed at any time in any place where there is available Internet connection [2].

2.2 Difference between IPTV and OTT

Both IPTV and OTT share the same fundamental principle of delivering multimedia streams via the Internet, but there are many characteristics that differentiate each service from the other one. The most important feature is that IPTV has a higher QoS and QoE because it is usually a more expensive service with higher priority service delivery unlike OTT which is commonly low cost service with best effort service delivery.

IPTV needs a dedicated server subscription and proprietary user receiver to get the service. However, OTT does not need any special or proprietary devices. IPTV is delivering TV broadcast over the Internet using managed networks. Meanwhile, OTT is delivering multimedia over the web using an unregulated network. There are also other detailed differences between OTT and IPTV. One of these differences is the common protocol used in each service which is HTTP over TCP for OTT and RTP over UDP for IPTV [3].

2.3 IPTV vs Traditional TV

Traditional television is a single direction transmission and broadcasting system that sends all channels at the same time to the users to choose from the available programs. IPTV is a double directional transmission multicasting system that streams only the requested program for each user. IPTV can provide a higher quality multimedia over the regular analog and digital TV because of better techniques used and higher standards including good compression of videos that reduces file size and preserves image quality [2], [4].

A combination of IPTV, VoIP, and Internet access is a convenient and compact way to provide these three services using the same end user modem [2]. On the other hand, conventional TV needs to have a dedicated cable line or medium. The minimum requirement of IPTV is that the bandwidth has to be larger than the sampling rate of the encoded video stream to get a smooth watching experience otherwise many packets can be delayed or lost which causes lateness and stops in displaying the video stream with larger impact on highly compressed video streams [5], [6].

2.4 Video on Demand

Usage of Internet data traffic is increasing rapidly, especially video streaming due to more access to smartphones and new devices connected to the World Wide Web together with the availability of faster mobile network connection as in 4G. 60 percent of total mobile data traffic in 2016 is mobile video traffic, and it is predicted to be 78 percent of the total mobile data traffic by 2021 [7]. About 40% growth in media streaming traffic each year is observed [8]. YouTube is one of the largest video streaming websites in the world where on average about 1 billion videos are watched daily [9]. Video on Demand is a popular and modern service that allows the user to play a video at any time. YouTube is the best example of a website which provides this kind of service. YouTube has a feature that its private users can upload and broadcast video contents for free, which makes this service popular.

Network congestion can become a significant issue that must be resolved when the network needs to transmit 1 billion videos per day. That is a gigantic number which come

from the enormous amount of devices connected to the network and has access to this service.

According to a Nordic study by Harrie [10], the usage of VOD viewing in the Nordic countries has increased. YouTube and Netflix are at the top of VOD services on these countries, and more than 50 percent of the people in Sweden watch YouTube weekly. In terms of daily watching, it is found that a third of the population of Sweden are watching YouTube daily.

2.5 Subscription Video on Demand

Subscription Video on Demand (SVOD) is a TV/film paid service where the user has to subscribe to this streaming service to be able to get access to a large amount of video content. It is mentioned in [10] that in 2015, young people living in the Nordic countries are seen to spend the majority of their time on smartphones and computers. It is also shown how the SVOD had rapidly increased, and the total consumer revenues have quickly risen from 40.7 million EUR in 2011 to 844 million EUR in 2014, which is about twenty times increment in four years.

In terms of how fast the usage of SVOD services has grown worldwide, a study [11] shows how the increasing of this type of service occurs. It is shown in Figure 2.1 that the home subscribers increased from 20 million to 117 million in year 2015, and the value is expected to be 249 million in year 2020.

The rapid growth in SVOD service usage leads to a gigantic increase in data traffic, which can cause network congestion. There are many reasons that make users subscribe to

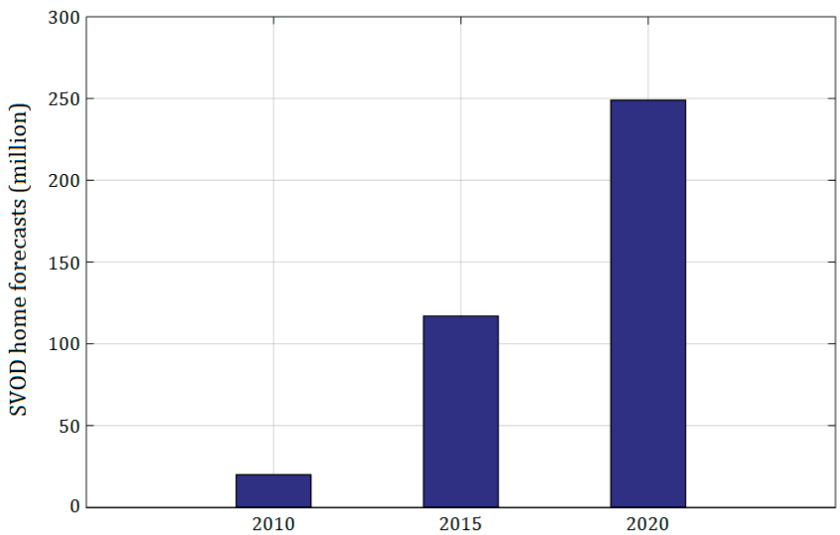

Figure 2.1: Previous, Current, and Expected Global growing of SVOD 2010-2020

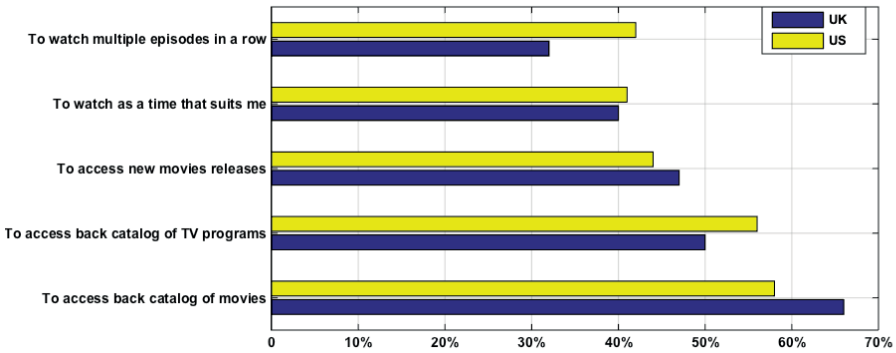

Figure 2.2: Top 5 Reasons that US and UK SVOD Subscribers signed up for, March 2014

SVOD service. Figure 2.2 shows the percentage of the reason to make people choose this type of service. This study is based on US and UK SVOD subscribers. The preferences of the users seem to be similar in both countries and most of the users have signed up for SVOD services to access back catalog movies or TV programs, i.e. to access Catch-Up TV service [12].

The relationship between the rapidly growing SVOD service and the data traffic is directly proportional. The huge increment in the data traffic can lead to network congestion due to large amounts of data transfer that the current network construction might not tolerate.

2.6 Catch up TV or Time shifted TV

An important and popular key feature of Video on Demand service is Catch-up TV, which is the ability to view previous episodes of a TV show or already started programs at any time. A recorded TV show can be seen if it was broadcasted less than a day or several days. The length of the video that is possible to be retrieved depends on the determined available storage assigned for recording the video stream in the server.

According to a study by Grece et al. [13], Catch-up TV service is the most widespread service among the available categories of VOD services in the EU. In Figure 2.3, Catch-up TV service comprises 33% of total available VOD services in the EU thus put this service in the first place.

Many TV programs become easily accessible online through a Catch-up TV service at any time. According to a report [14], about 64% of Free to Air TV program broadcast aired at evening time in March 2010 are already available online. The number of available TV programs keeps increasing each year as it was 59% in September 2009 and 53% in March 2009. Note that each content is available under one week, then it was replaced with a new content. It is also mentioned in a later report [15] that about 60% of the Internet users used Catch-up TV in 2011 compared to 54% in 2010.

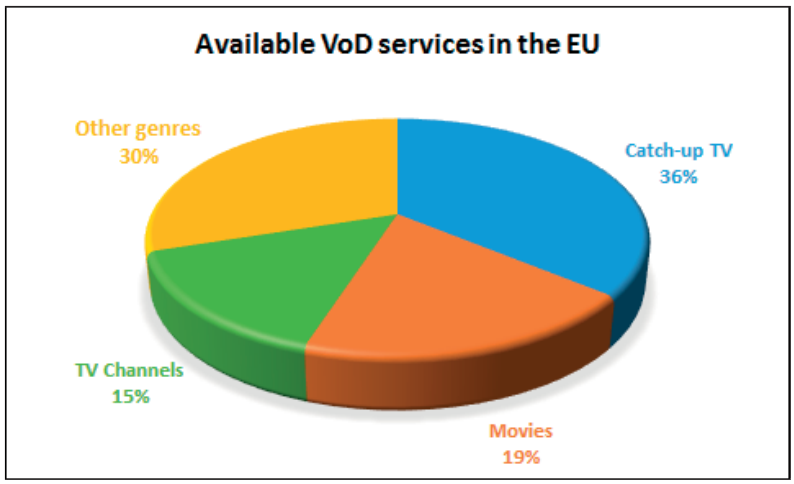

Figure 2.3: On-demand available audiovisual services in the EU

2.7 Streaming and Peer to Peer sharing

Peer to Peer (P2P) is a data sharing method that involves two or more users that cooperatively shares portions of data directly from one user to another one without the need for a central server. P2PTV is a type of data sharing that is sharing the video stream watched by a user with other users by simultaneously download and upload the video stream.

With video streaming, user can watch the video directly without the need to wait until the entire video file is downloaded. The video can be watched directly while the bit stream of the content is being downloaded. P2P video streaming adds the opportunity to upload the already downloaded and watched part of the video by users and share it with other users to watch it. More connected peers (users) improve the availability of the video file and enhance download/upload speed for peers that is collecting file data from many sources at the same time. The sources are other users instead of a main server. P2P techniques will effectively decrease the traffic load on the main server which will minimize network congestion, increase the reliability of the network, and reduce the cost of the system [16].

2.8 Prefetching and Caching

It is mentioned before that zapping time is an issue that should be taken in consideration when talking about video streaming. Many potential solutions can be implemented to reduce long zapping time. Prefetching and caching are some of the methods that can be used to reduce the start-up delay, and the buffering time, which is the time measured from the request for a content until it is ready to be played at the end terminal. Prefetching is actively predicting and pre-loading the next expected content based on the previously seen contents. On the other hand, prefetching would effectively reduce the buffering delay from the first request.

In an article by Du et al. [17], the researchers studied and analyzed prefetching

schemes, infrastructure of a prefetching system and video prefetching selection methods. In the context of prefetching amount, this is when it is too much amount associated with network congestion through increased data traffic. Prefetching has been used in YouTube service to predict the next video that the user may attempt to watch.

In order to understand the prefetching system we have to know a lot of details about its infrastructure. The system comprises of two engines: the prediction engine and prefetching engine. The prediction engine has the task to predict what users are going to watch next time before they click the content. The prefetching engine has the task to determine what content to prefetch [17], [18].

Caching is passively saving the requested content in a cache to be used for reducing the latency when the content is requested again. Caching is working together with prefetching to reduce latency and data traffic. In order to reduce data traffic, caching is a solution that can be implemented at the end client or at the proxy server [19].

Many algorithms are used in caching system to remove unnecessary content stored in cache to free up memory and disk space. Least Recently Used (LRU) is one algorithm which is used to remove content that has been in the cache for a long time, while the timer for each content is reset when it is used [20]. Another algorithm which has been used in caching is Least Frequently Used (LFU), which removes the content that has the lowest number of requests for download. Adaptive Replacement Cache (ARC) is a recent algorithm that combines the two previous algorithms and consider both frequency and time of use when removing content from the cache. ARC has a better performance than the other two algorithms [19].

The relationship between the cache size and the average zapping time, using any caching strategy, is inversely proportional. It seems that the zapping time is decreasing when the cache size increases because more data can be stored in high speed caches close to the users [21], therefore increasing cache size is a simple method to reduce zapping time. However, cache memory is an expensive resource and it is not affordable to make it too large, but still video streaming service needs a cache size larger than the traditional text based web service [22].

2.9 Behavior of Users

It is important to study user behavior in order to see the effect of user access pattern and user zapping rate on the network. Usually, users do not change their viewing habits, therefore this habit can be exploited to find a good prefetching and caching strategy in order to optimize network infrastructure and management processes to enhance Internet based TV services.

User behavior had been studied in several papers in different ways. For example, in [1], [23] and [24], user behavior had been studied by analyzing IPTV access traffic to check the influence of user access rate on the network. The first and second study were based on a Swedish VOD service, whereas the third study was based on a Chinese VOD service at a different time. However, all papers have similar conclusions.

The papers studied user activity in different aspects. In the first paper by G. Yu et al. [1], they measured zapping rate per minute normalized by the number of active hosts at different times of the day. In the second paper by A. Ali Eldin et al. [23], they analyzed the distribution of request arrival rate and observed Zapper user behavior. In the third paper

by H. Yu et al. [24], user arrival rate is measured, and user access pattern is plotted across the time of the day. Although, these papers used different methods to study user activity, the users show similar behavior with respect to the time of the day, i.e. similar shape of hourly access pattern in the day for user arrival rate and zapping rate. The average number of active hosts and zapping behavior, increase at the evening time and decrease at the day time. The top value of the highest number of active users and also the highest number of Zappers occurred commonly at approximately 20:00 o'clock [1].

2.9.1 Zappers

One kind of user behavior which is a widely known watching habit is to skim through many channels searching for an interesting program to watch. These users are called Zappers who have less patience than normal or Loyal users who keep watching TV programs to the last minute, and occasionally change the channel. Zappers can be a problem for VOD servers because of the increase of load on servers due to the Zappers' requests. Many video streams should be cached to decrease the latency in order to enable a smooth watching experience and a fast respond to users' request for any possible expected video to be watched by the users. The uncompleted video download is a waste of bandwidth and network resources since a large portion or maybe a full stream of video is sent to the user but discarded and not actually used by that user [25].

The benefit from studying the effect of Zappers on the network is that it concludes the need for a better caching strategy which has to be done to reduce the negative influence of Zappers on network resources. In both of the two studies [23] and [24], about half of the users have terminated their current streaming sessions before 12 minutes. Depending on the duration of streaming session, users are sorted as Loyals or Zappers, considering users having a shorter session time as Zappers. In Figure 2.4 session termination time is shown according to this study [23] about the VOD workload analysis of the Swedish TV4 channel. These kind of statistical studies are important to do in order to mitigate the negative effect of Zappers on server resources by trying to enhance the prefetching and caching processes in VOD networks.

2.9.2 Loyals

The other kind of user behavior is Loyal users who usually do not change the channel often and tend to completely watch their favorite programs. This kind of users is better from the server point of view because they add less load on the server compared to Zappers user behavior. Zappers need larger cache sizes and more bandwidth since they change the channel often. Loyals in the other hand, need a low cache size and a small bandwidth because the download time of the video stream is short and can be spread over time without compromising video quality and QoE for users. In the case of Zapper users, the network is usually over-loaded with requests, and the bandwidth is fully loaded all the time. Number of requests for a content is lower in the case of Loyal users, unlike Zapper users who have a high number of requests for contents which have to be cached in every request made by the user.

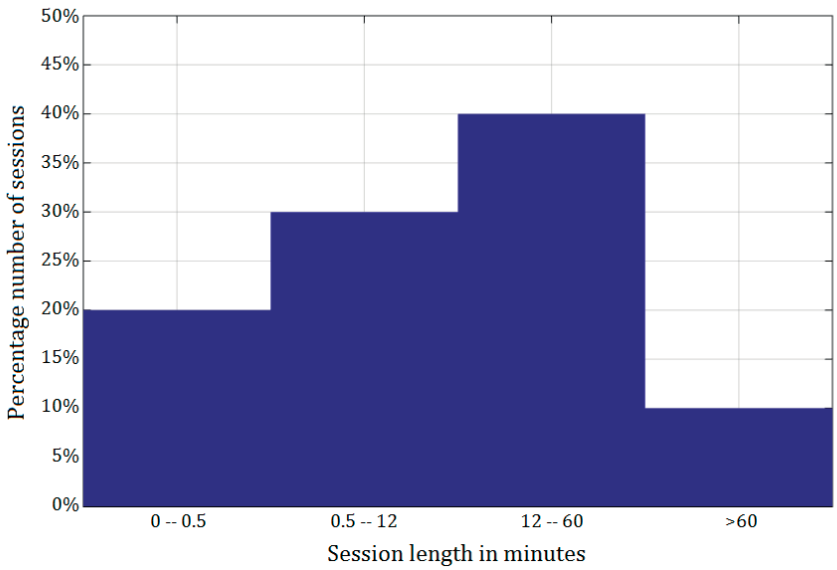

Figure 2.4: Distribution of session length

Chapter 3
Methodology

In this chapter, we will demonstrate the methods we used to analyze the data set. The main tool used to inspect the data is Python. The reason of preferring Python over Matlab is explained by presenting the differences between these two computer programs. We will also draw and describe the construction design of the prefetching system, and present our idea about the feasible caching strategies. Each part of the system structure will be described in detail.

3.1 Python vs. Matlab

The analysis tool used in this project is Python v3.5.1 through Spyder 2.3.8 on Microsoft Windows and Mac OS. We found that this version is the best one for running our data analysis codes in Python within a reasonable time. The processing time for 1 month of data was about 10 minutes or less.

After trying other newer versions of Python, we found that it may not be fully compatible with our code because of some changes in the updated program that cause delays and crashes which prevent us from getting any results. Therefore, we used the mentioned version of Python, which was the most suitable one for us.

We had two options about which computer program to use in data analysis, Python or Matlab. We preferred Python over Matlab in programming all codes used in processing the data in this project because it is easier to handle the data in Python than in Matlab and there are also many advantages in Python over Matlab, like the portability of the codes, which can be run on any operating system [26].

Python is an open source computer program which uses a dynamic object-oriented programming language and has many algorithms and packages which are free to use. Matlab is a commercial program which is not free and has proprietary algorithms that cannot show the actual source code. Python is more powerful in processing a large amount of data with simple and compact codes, and it can show the result in a really short time. Many libraries, packages, modules, and toolkits are available for free to use in Python. Some of these libraries used in this project are, Numpy, Matplotlib, and Pandas.

NumPy (Numeric Python) is a Python module used for common mathematical computation of numerical data in arrays and matrices. Python program core in combination with its libraries like Numpy and other modules, can act just like Matlab program functionality [27].

Matplotlib is a library package in Python used as a script to generate 2D plots, diagrams, charts, and histograms. Many statistical graphs in this project produced using matplotlib in Python [28].

Pandas is a fast and flexible toolkit in Python that uses low computer memory in analyzing large data. Pandas is very suitable for dealing with information imported from SQL database and saved as a text file, which can be loaded as a data table and categorized into columns and rows in an array of DataFrame [29].

Pandas can easily handle and remove missing data (NaN) which is the case we have faced in our data that some fields for a few users were blank. Less than 1% of users' data were missing so we have safely ignored these specific uncompleted data for these users because it will not significantly affect the result of the data analysis.

3.2 Prefetching System

The main idea of this thesis is to define a prefetching system that can handle the increasing demand on the Internet based TV service. The access log on the server for the service users, is stored in a database. The registered information about user requests and content details can be analyzed to determine user behavior, content length and popularity. The cache setting and the network configuration are set based on the obtained values from analyzing the gathered information about the user activity and the content properties.

The prefetching system has a prediction engine that predicts the probable contents that may be viewed by each user later, and also concludes the best setting that can be used to enhance QoS and QoE. Many goals can be achieved by choosing the best-predicted settings for cache and network configuration. The main goal is trying to solve the major issues that faces the reliability of the network and the service, which are network congestion and time delay. The prediction system can also reduce the latency problem caused by Zappers by providing them a more variety of content with less caching. The ultimate goal of the system is to deliver high quality videos smoothly to all users.

This section gives an overview of this thesis work. An outline of the studied parts in the following chapters is shown in a single Figure 3.1 using a flowchart. The prefetching engine is represented in this Figure, which shapes the main frame of the prefetching system that actively decides which content and how much of it should be stored in the cache. This will help in eliminating unnecessary data storage and transfer. Another important part of the prefetching system is the prediction engine which tells the system what a user may plan to watch next based on previously watched contents [17], [30].

Figure 3.1 illustrates a scheme of the prefetching system using simple blocks in a flowchart to show its mechanism. This figure also shows that the prefetching system takes the needed information from the previously registered data in the database about the usage of the IPTV service. The result of the prefetching engine decision is applied to the cache of the server or user depending on the predicted settings based on analyzing the collected information from the database.

There are many branches in the flowchart of the prefetching system. Each branch represents one studied section using one type of data analysis. All branches start from the database by collecting log data for one user or all users depending on which data analysis is conducted. The result from all the branches is cooperatively affecting the settings of the cache for the server or the user. The first and the second branches on the left side of

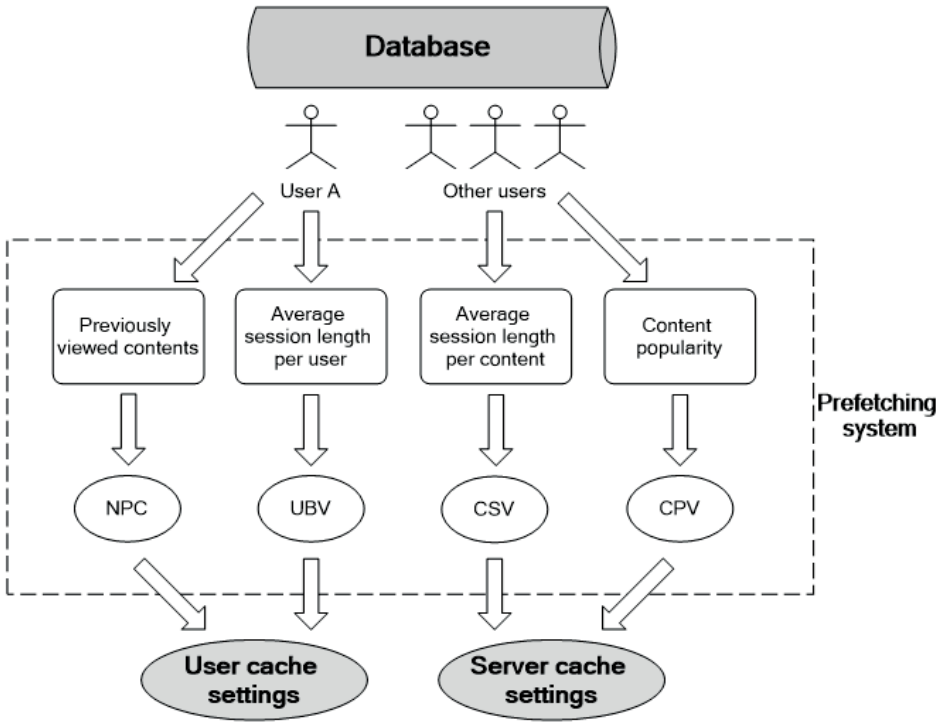

Figure 3.1: Prefetching system chart

the chart are more directly related to the current user (User A) and its result will affect the cache of that user. The other two branches on the right side of the chart take the data requests of all users into consideration and analyze them to give values that can be used to setup settings and configuration of the server cache and its parameter.

In the following chapters, we are going describe each branch in a separate section. In this section, a short summary of each branch is introduced.

The first branch in the prefetching system is the predicting engine that analyzes the information about the previously viewed contents for each user, and predicts the Next Possible Content (NPC) that may be viewed by the user, which can be a list of other similar videos or one video that is the most possible content to be viewed by the user i.e. the next episode of a series that the user has just watched. It can also give a single value of how much percent of the content has to be cached based on the probability of viewing that content. It will also add more reliability to the prediction system, make its results more relevant to each user, and set the cache with the correct setting.

The second branch is about the estimation of user behavior from the average session length for each user. The result is given as a value for each user that determines if that user is a Loyal or Zapper based on the User Behavior Value (UBV). Since this value indicates the user activity, it can be used to set the user cache, and to choose a suitable system settings and correspondent network configurations in order to improve QoE for users and QoS for the server. For example, by setting a small size of each cached content, but at the same time, many contents are cached to reduce the delay time of fast browsing among

many contents to enhance the experience of Zapper users. For Loyal users, the opposite procedure can be used to improve their experience by increasing the size of the cached content and caching only the most expected contents to be seen by Loyal users.

The third branch is about calculating the average session length per content and getting Content Session Value (CSV), which is a value that counts the average seen part length of each content by any user. This value helps the prefetching system to check the percentage of any content that has been watched and then decides how much of that content should be cached into the server cache. The size of cached content will be higher if the average session length of that content is high.

The fourth branch is about checking the popularity of contents and sorting all available contents according to the rank of each content. The popularity of contents is found by counting the number of requests for each content that is registered in the database during the studied time period. The result of content popularity gives a Content Popularity Value (CPV) for each content. The probability of caching a content and the percentage size of that cached content increase with the increase of CPV. Thus, the top ranked video content has a high chance to be already loaded in the server cache with large portions of that video.

Chapter 4

Data Analysis

In this chapter, we will introduce the analysis process of the data set, and show a sample of the obtained data. A description of the data set is expressed, and statistics of the data set are shown. The data analysis is done in many aspects. The effecting factors are explored, and each factor is inspected in a separate section. These factors are the content length, content popularity, session length, and user behavior. We made equations to evaluate some values, which can be used to identify the user behavior and content properties. The prefetching and caching system are explored, and the appropriate cache setting is explained. We also remarked the relationship between all these criteria.

4.1 Dataset

In this project, a data set from a Portuguese Catch-up TV service provider is analyzed to study user behavior in order to enhance the performance of On Demand TV service and network. Depending on user behavior a proper caching and prefetching scheme can be designed to improve the performance of the system.

The data set contains logs of information about On-Demand service usage between the third of June 2014 until the end of June 2014. About 16 million user requests are registered during this period of time, and about 400.000 programs are viewed by over 500.000 subscribed users within the Portuguese On-Demand TV service provider.

The needed information from the database is exported and analyzed with Python scripts to get statistical graphs. The result of the analyzed data is retrieved from the plotted graphs that show user activity, session length, and content popularity during the time period of the collected data.

Table 4.1 shows a sample of the data set used in this project. The data set contents consisted of nine columns and many rows, where each row represents one request for a program or a video content. The first information in the data set is account id which represents user id anonymously in a set of unique letters and numbers for each user. Another provided information in the data set is playtime which represents the request time of the content or episode, content's start time, content's end time, content ID, and other information about the content.

Table 4.2 shows statistics of the data set which consists of a log of user requests for one month. Total sessions correspond to the total number of valid requests after removing invalid requests which have empty fields in the data set. Non-unique contents count all

Table 4.1: Sample of the data set

Account	PlayTimeHour	Start	End	EpgPID	EpgID	Se.Nr.	Ep.Nr.	CallLetter
00005543E3F016079E5DDCC88F111095F	2014-06-04 16:12:44.697000000	2014-06-02 20:19:00	2014-06-02 20:45:00	6371050	29203	5	1	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-04 16:44:30.407000000	2014-06-03 20:19:00	2014-06-03 20:45:00	6373735	29203	5	2	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-06 21:57:45.943000000	2014-06-06 00:48:00	2014-06-06 01:12:00	6373735	29203	5	2	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-06 22:22:12.673000000	2014-06-04 20:19:00	2014-06-04 20:45:00	6375644	29203	5	3	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-06 22:49:17.957000000	2014-06-04 20:19:00	2014-06-04 20:45:00	6375644	29203	5	3	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-06 22:49:23.497000000	2014-06-05 20:19:00	2014-06-05 20:45:00	6378680	29203	5	4	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-07 21:36:10.297000000	2014-06-07 10:32:00	2014-06-07 10:55:00	6378680	29203	5	4	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-07 21:47:16.283000000	2014-06-07 10:32:00	2014-06-07 10:55:00	6378680	29203	5	4	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-07 21:47:28.013000000	2014-06-07 10:55:00	2014-06-07 11:16:00	6381980	29203	5	5	FLIFE
00005543E3F016079E5DDCC88F111095F	2014-06-14 13:23:41.827000000	2014-06-13 00:20:00	2014-06-13 02:10:00	6401145				HOLLW
00007B30D235FDBA4131DE3459DC2F84	2014-06-03 23:20:19.333000000	2014-06-01 20:17:00	2014-06-01 21:10:00	6388579	29238		1	TVI
00007B30D235FDBA4131DE3459DC2F84	2014-06-04 00:16:09.980000000	2014-06-02 22:48:00	2014-06-02 23:35:00	6373407	29238		2	TVI
00007B30D235FDBA4131DE3459DC2F84	2014-06-04 23:33:07.473000000	2014-06-04 20:45:00	2014-06-04 21:55:00	6378410	24839		203	SIC
00007B30D235FDBA4131DE3459DC2F84	2014-06-05 00:36:42.203000000	2014-06-04 22:58:00	2014-06-05 00:04:00	6372934	27843	2	10	FOXHD
00007B30D235FDBA4131DE3459DC2F84	2014-06-05 01:29:33.683000000	2014-06-02 22:56:00	2014-06-02 23:46:00	6376489	28287	2	7	AXNBL
00007B30D235FDBA4131DE3459DC2F84	2014-06-07 00:38:38.250000000	2014-06-05 20:37:00	2014-06-05 21:46:00	6378426	26944		95	TVI
00007B30D235FDBA4131DE3459DC2F84	2014-06-07 00:38:38.597000000	2014-06-05 20:37:00	2014-06-05 21:46:00	6378426	26944		95	TVI
00007B30D235FDBA4131DE3459DC2F84	2014-06-07 01:31:46.847000000	2014-06-06 20:37:00	2014-06-06 21:45:00	6381786	26944		96	TVI
00007B30D235FDBA4131DE3459DC2F84	2014-06-07 01:31:47.033000000	2014-06-06 20:37:00	2014-06-06 21:45:00	6381786	26944		96	TVI
00007B30D235FDBA4131DE3459DC2F84	2014-06-10 21:37:13.940000000	2014-06-08 14:23:00	2014-06-08 15:15:00	6388535	26644	2	20	RTP1

Table 4.2: Statistics of the data set

Total no. of sessions	Non-unique contents	Unique contents	Total no. of users	Non-active users
15 871 557	46 544	35 845	564 771	42 410

the available contents and its copies with the same EpgPID. Unique contents count only contents with unique EpgPID excluding all copies.

Total users count all users with unique account ID including active and non-active users. Non-active users count only single request users that have only one request in the entire data set.

4.2 Content Length

The available video content length varies depending on the type or the genre of the program, and can be from a few minutes to several hours. In our data, the minimum video length is 3 minutes, and the maximum video length is over 120 minutes. Over 35000 videos have been requested by the subscribers from the server of the service during the captured user request log period.

In our calculations, we have found that for some programs, there are slightly different content lengths with the same program ID (EpgPID). We also noticed that when the program is broadcasted in the morning it is a little bit shorter than if it has been broadcasted in the evening. This content length non consistency is possibly occurring due to the inclusion of different advertisements length during the program broadcasting. Hence, some programs are shorter in the morning than in the evening for the same EpgPID because the ads are shorter in the morning due to lower number of viewers in the morning than in the evening.

If we encounter differences in the content length calculations, we consider only the length of the program that appears first in the list of the given data. This slight difference in the length of some contents will not affect the result of data analysis too much because it is a short amount of time difference compared to the whole length of the program. If we consider the same program with different length as a different program, this will increase the number of available programs which can noticeably affect our data analysis result.

4.3 Content Popularity

In this section, content popularity, or content rank is investigated. Content popularity is defined by the total number of requests for a content, which is widely varied among the available On-Demand videos. Content popularity depends on the type, or the genre of the program, and if it is a new or an old program. The popularity of some programs drops

dramatically when it gets too old, but some programs become more popular with time, and some programs maintain its popularity.

There is a vast difference between the popularity of some contents compared to other contents. Only a small number of videos have a huge popularity with thousand of requests every day, but the rest of the videos have an extremely low popularity. A large number of videos have got only one request in the whole time period of the data set.

It is important to study content popularity, because it helps network developers to distribute network resources in a better way, and also helps the network designer in deciding the proper setting of system parameters, such as cache size and bandwidth. This is done by analyzing the collected information about user requests on video contents. The result gives an image about which contents have high rank, and which contents have low rank.

Network resources are considered as a limitation factor to have a reasonable quality of service and quality of experience. Therefore, it should be distributed carefully among users to serve them with the requested content. High ranked videos should get high network resources, but low ranked videos would get low network resources, but enough to be fairly viewable by the users. However, abandoned videos with no request in a long period of time should not occupy network resources, that might take a large cache memory space in the server. Although, they could be kept in the server storage for a while, then they can be removed later.

Content Popularity Value (CPV) is a value that we developed as one of the grading scales to the contents. It decides how much cache size should be assigned to each content based on its popularity. This value is directly related to content rank and can be calculated by counting the number of requests for the content. Then the result can be scaled to a value between 0% and 100%, where 0% is denoted to abandoned videos, which will be removed from the server cache, while 100% is denoted to high ranked videos, which will get high priority and large space in the server cache.

The initial value of CPV for new contents can be set as 50%. This is the case when a video with an unknown popularity gets a new entry in the available On-Demand video list. Then after a period of time, that value can be changed according to the number of hits on that new video. However, this is not a perfect assumption if the new video is expected to be very popular based on its initial recensions, or if it was a sequel to another popular video. Therefore, the provider can decide to change the initial values for some videos according to his own algorithm which is based on different expected popularities to the new videos.

4.4 Session Length

Session Length is the actual viewing time of the currently requested video by a user. Session length for each specific video request can be equal or less than the time length of that specifically required video.

A significant limitation that we found in the data set is that there is no registered information available on session end-time. We do not have any trace if a user has ended the session and there is no timeout for user connections and we do not know whether the user has completely seen the requested program or just a part of it. This makes it difficult to take a decision about at which time the user finished watching the video or the program. Therefore, we have to assume the end time for each viewing session in order to do our

data analysis about user behavior.

It is important to note that the session end-time for a user is limited by the start-time of the next session for that user. This belief is based on a presumption in IPTV systems that each unique user account is not allowed to open more than one session at the same time. The user may stop the current session and open another one by requesting another video. This new request will terminate the current video stream and start a new one.

4.5 Session Length Calculation Assumptions

In order to begin the analysis of the data and to draw graphs showing user behavior, each session length in the request list has to be determined. Since there is no information available about the end-time of a session, each session end-time should be estimated reasonably based on the time of the next request for a user. However, we should always take care that the session length cannot be larger than the video content length in our assumptions, hence session length for a content request is limited by the length of that content.

By checking playtime for the requests of a user, we could notice that there are a variety of differences between these playtimes. At first, we could simply assume that each session length is equal to the difference between two consecutive request playtimes of a user. However, the playtime of the next request may exceed the actual video content length, which is the maximum limit for the session length of that specific request for the designated video content, i.e. the session length is equal or less than the content length.

Session length has to be determined when the difference between two consecutive playtimes requested by a user is larger than the content length of the requested video by that user. In this case, for example, we can try one of three options to determine the session length for the current requested content. The first option is to consider that the user has watched the whole designated video content, i.e. 100% of the video content is viewed. The second option is to assume that the user has watched half of designated video content, i.e. only 50% of the video content is viewed. The third option is to assume that the user has not watched that designated video content, i.e. 0% of the video content is viewed.

In the result chapter, we will take the first option only in our calculations because it will take too much time to discuss all the three options. However, we also tried to check the result of the other two options. We noticed that the result is shifted towards the assumed ratio, which we have taken in our calculations of session length. The assumed ratio can be 100%, 50%, 0%, or any other possible value. In our calculations, we took only the first option with a ratio of 100% because we think it is more likely that the user would continue viewing the content and finish it to the end, if that user did not cancel the session, already in the beginning. However, the first option of assuming that the users tend to see the whole length of the requested videos, leads to shift the result of user behavior more towards the Loyal side instead of the Zapper side. This will be discussed more in User Behavior Section 4.6.

On the other hand, session length is assumed to be equal to the difference between two consecutive playtimes of a user, if the difference is equal or less than the requested video content length. i.e. percentage session length can be anything between 0% and 100% of the content length.

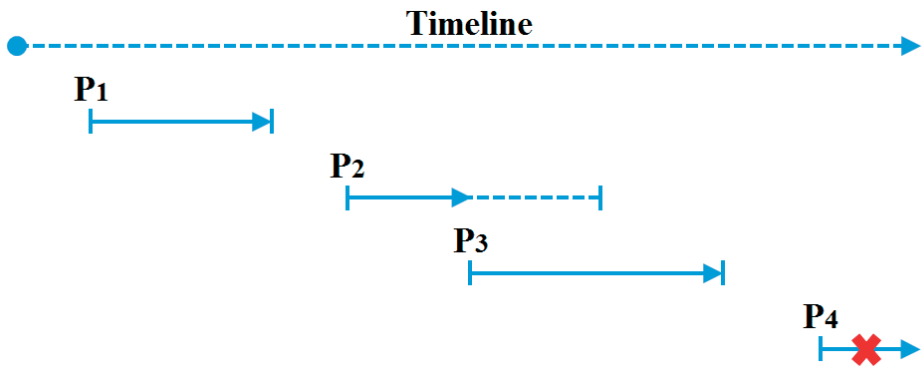

Figure 4.1: Sketch of session length calculation

An example of session length calculation is explained in a sketch in Figure 4.1. The timeline of the data log is plotted as a long dashed line. The blue dot on the left of this line is the start point of the data log period, and the end point is on the right. In this example, four requests by one user are shown, where P1 denotes the playtime of the first request and P2 denotes the playtime of the second request, etc. The length of the requested content is represented by the distance between two vertical bars on the same line in the sketch. The left bar is located at the playtime of the request, and the right bar is the end-time of the content. The arrow lines represent the session length, where the end-time of the session is located at the right side of the arrow.

For the first session when P2 comes after the end-time of the first content, it is assumed that the user has watched the whole video. For the second session when P3 comes within the duration of the second content, it is assumed that the user has terminated the second session at the same point of the next request playtime P3. It is visible in the graph that we have removed the last session of this user, and also all users, since the next request playtime is located out of the data log timeline.

The following Equations 4.1 and 4.2 express the previously mentioned method in calculating session length. In order to calculate the average session length per content, the sum of all sessions for each content is taken, the result is then divided by number of sessions for that content in the data set. The average session length in minutes is found using Equation 4.3, and in percent using Equation 4.4. Note that the average session length in percent, which is named as Content Session Value (CSV) is one of the factors that we used to determine cache settings. This value represents the average session length for each content, and can be used to decide how much of each content should be cached.

$$S_i[min] = \begin{cases} C_i & \text{if } P_{i+1} - P_i > 1 \\ P_{i+1} - P_i & \text{otherwise} \end{cases} \tag{4.1}$$

$$S_i[\%] = \begin{cases} 1 & \text{if } \frac{P_{i+1}-P_i}{C_i} > 1 \\ \frac{P_{i+1}-P_i}{C_i} & \text{otherwise} \end{cases} \tag{4.2}$$

$$S_{avg}[min] = \frac{\sum_1^m S_i[min]}{m-1} \tag{4.3}$$

$$S_{avg}[\%] = \frac{\sum_1^m S_i[\%]}{m-1} = CSV \tag{4.4}$$

S_i is the session length for request i
C_i is the content length for request i
P_i is the playtime of request i
i is the index of a request for a content where $1 < i < m$
m is the number of requests for a content
CSV is the average content session value

4.6 User Behavior

Studying user behavior is important to understand On-Demand video service usage and get the needed information to improve system configurations by selecting the proper size of caches in order to enhance QoS and QoE. User behavior is how each user acts when using the IPTV service. User requests to the IPTV service should be analyzed in order to identify user behavior and to characterize user types.

There are two main user groups that can be defined according to their average watched session length, which is measured by using their request activity and based on session length calculation assumptions. These two groups are Loyal users and Zappers users. Loyal users tend to completely watch TV programs to the end or most of its parts, and they seldom change the channel. Zappers users are more impatient in watching long videos, and they change the channel more often and try to browse among different videos.

Many methods can be used to model user behavior. We tried to find a simple way to identify user behavior in our data set. Our method, which we used to calculate user behavior is to take the average of the length or the percentage of the watched contents for each user and use this result in calculating the User Behavior Value (UBV), and then we compare the results to decide if the user is a Loyal or a Zapper. The result of the user activity estimation is plotted in two graphs, one for the calculation in minutes and the other one for the percentage calculation. These graphs will be provided in Results Chapter 5.

The following Equations 4.5 and 4.6 are used to calculate the User Behavior Value (UBV) by analyzing the information in the data set including user requests and content start/end time to find the average session length in percent. The equations are applied to the list of requests for each user by taking the average of percentage sessions viewed by each user over the number of sessions for that user. The result of the Equation 4.6 is the percentage average session length per user, or the so-called User Behavior Value (UBV), which is needed to decide if the user is a Zapper or a Loyal.

$$S_j[\%] = \begin{cases} 1 & \text{if } \frac{P_{j+1}-P_j}{C_j} > 1 \\ \frac{P_{j+1}-P_j}{C_j} & \text{otherwise} \end{cases} \tag{4.5}$$

$$S_{avg} = \frac{\sum_1^n S_j}{n-1} = UBV \tag{4.6}$$

S_j is the session length for request j
C_j is the content length for request j

P_j is the playtime of request j
j is the index of a request for a content where $1 < j < n$
n is the number of requests for a user
UBV is the average user behavior value

The purpose of calculating the user behavior value is to find an average value that can represent the watching behavior of each user. This value is calculated by normalizing the average length of sessions for each user. The resulted user behavior value can be equal to any rate between 0% and 100%, where 0% is Zapper user behavior, and 100% is Loyal user behavior. We also added a negative percentage value in the graph to represent inactive users. That negative value counts all users who appear in the data set with only one request during the whole data log time period. If a unique user account is listed only once in the data set, we should make a decision if that inactive user will be listed with other active users or not. We decided to add these inactive users in a separate bar in user behavior graph 5.13, which will be presented later in Results Chapter, to compare the total number of inactive users with the other active users.

We decided to not combine the users who have made only one request with the other users, because we think that it is not reasonable to give much attention to these users that used the service only once in a whole month of IPTV service usage. Another reason is that it adds more complexity to the result and adds more assumptions, which are if those inactive users have fully, partially or never seen the videos that they had requested. Therefore, we assumed that these users did not watch the requested content. However, it is better to enclose these single request into one tab in the user activity percentage graph to try not to affect other information on other users with many requests. This negative value shows how many single requests were encountered in the given data set.

On the other hand, for active users with many requests listed in the data set, an issue similar to inactive users is always emerged in calculation of user activity. This issue is about the last request of each user in the logged data set time period when there is no next request is registered in the data set, because it is out of the data set time period range. We also made the same assumption that all users have never seen the video content related to their last registered request because the request that comes after is out of the data set time period range. Therefore the last request for each user is only used to calculate the session length for the last requested content.

In those two cases of inactive users and last user request, we assumed that the users did not watch the last requested video. However, we can also assume the opposite and consider that the users have fully seen the last requested video content and add these sessions in the calculation of user behavior value. In this case, the result will be shifted more to the Loyal user side. Therefore, the result will not be logic and will be more biased by adding the last request of each user as a 100% session length to the sum of previously viewed sessions for each user.

4.7 Prefetching

The prefetching system is described in Section 2.8 where it was divided according to the task of the predicting and the prefetching engine. In this project, we have focused and investigated the prefetching engine using a multiple branches algorithm as shown previously in Figure 3.1.

The predicting mechanism is not used to suggest contents to users, but it predicts the next possible content which the user may prefer to watch. The prefetching mechanism is a tool to decide the amount of the predicted content that should be buffered to the cache then over the network to the user. Understanding user preferences, sessions, and contents, are important paradigms for this research work to enhance the prefetching algorithms in order to specify the right content to prepare and to determine the right amount of data ready for buffering process. Data analyzing is the provided tool to study and describe all of these concepts which can be shown in graphs to have a better view of the result.

In this section, we describe an example of what we have obtained from the data analysis and its benefits, and also how to connect the researched factors to increase the performance of the prefetching system. The data analysis process in this project has an aim to find a utility to increase the ability to prefetch the correct amount of data. It leads to a decrease of unnecessary data traffic. An example for the prefetching process, when the system wants to preload a content with a specific content popularity to two users, and they have different user behavior value, the amount of data should be different depending on these values.

Prefetching methods and its related background have already been discussed in Section 2.8. The relationship between the problem formulation and the prefetching perspective will be investigated later. Since we have to reduce data traffic and maintain quality of service level at the same time, we can preload parts of the predicted contents for the users. We have inspected the content length and popularity to understand the relationship between these criteria and the amount of prefetched data in the server. The location of the server has not been investigated. Instead, we aim to cache the content closer to users, or at the user's end terminal.

The benefits of the Prefetching engine are predicting contents which are going to be watched by users, prepare these contents to the caching system and trying to decrease the traffic on the network. A simple act that will reduce data traffic during rush-hour of the network is to preload the needed data to users during off-peak time when the network is not heavily used.

4.8 Caching

It is slightly mentioned in Section Prefetching and Caching 2.8 in Page 9 about different caching algorithms and what they depend on. These are the used algorithms in the current caching system. A crucial approach to have a better caching system is to improve the currently used algorithms. The effect of the improvement is to reduce delay time and data traffic in the system. The main goal of the data analysis is to find a simple way to enhance these algorithms. One of the important factors that affects caching algorithms is user behavior. The reason to study user behavior is to enhance those caching algorithms in order to reduce unnecessary data traffic and try to provide the ideal amount of data to users over the network. Session length calculating concept, per user and per content, has a wide effect on the analysis outcomes which determine the amount of data that will be implemented in cache. ARC algorithm has been studied as a caching algorithm which considers both frequency and time of use. User Behavior Value (UBV) has been used as a key value when we want to implement and improve ARC algorithm. The result of the caching algorithms is used to find the best way to manage the cache setting based on the

analyzed factors of user behavior, session length, content length, and content popularity.

4.9 Cache Setting

From each studied section we obtained three estimated values that can be used together to adjust the portion of cache size for each user and content. These three values are UBV, CSV, and CPV. These calculated values can be combined together and used to set the cache partition size, disk space, bandwidth or any resource assigned or used, for each user or content.

The prefetching factor PF is also another value that helps to enhance the adjustment of system settings according to user behavior which is the prediction of user behavior based on which type of content that each user are going to select.

All these values can be used collaboratively to obtain one final value that is used as a limit for each user or content to make a fair distribution of server resources. The weight of each value for a user/content is not always the same for all of those values. The weight of each value depends on the priority of each section. UBV concerns users only and not the content. Oppositely, CSV and CPV are only used for the content not the users. The final estimated value after calculating all those values in each section, can be linked to both the user itself and the requested content by that user.

$$SCS = a \cdot CSV + b \cdot CPV \tag{4.7}$$

$$UCS = c \cdot UBV + PF \tag{4.8}$$

SCS server cache setting value.
UCS user cache setting value.
CSV content session value for the given content.
CPV content popularity value for the given content.
UBV user behavior value for the user.
PF prefetching factor depends on user viewing history.
a, *b*, and *c* are fixed weighting constants that decide the weight of each specified term.

These two Equations 4.7 and 4.8 are not meant to be taken as a rule that gives determined values, but these two equations give an overview of which elements should be taken in consideration when determining system requirements and in deciding the best distribution strategy of usage of the server and the user cache.

The weight of each element or term is remarked in these two equations to have the opportunity to pay more attention to one term at the expense of the other terms, if we need to use unbalance terms, when some terms are considered more important for the system than the other terms. These weights can be manually adjusted by the service provider to have better control on giving the priority to some terms in the system settings adjustment.

These two equations can also be expanded by adding more terms when a new sort of analysis is conducted and more data types are collected, when it is necessary to give more accurate results.

SCS decides how server cache is managed and regulated, based on the service requirement, contents popularity, and average session length.

UCS decides how the user cache is managed and regulated based on user behavior, i.e. user session length, prefetching factor, and user viewing history.

UBV determines user behavior and show if the user is a Zapper or a Loyal. This value is found based on how much percent the user had seen the requested content. This value does not change much with respect with time for a user, but it can change slightly if the user changed his viewing behavior.

CSV shows the percentage of each content that is seen by users, and CPV shows the popularity or number of requests for each content. CSV is similar to UBV but the only difference is that CSV is related to the content, whereas UBV is related to the user. However, both CSV and UBV average values do not change that much in relation to time. On the other hand, CPV depends much on the time and it always increases with time, because the number of requests for a content will always increase with time. However, the rate of increment is not constant for CPV, i.e. it is more likely to have slow or medium increase in popularity of a content, in the beginning of its release date, then it gets a vast increase after some days. Later, after a period of time it begins to decrease in popularity at the end, when the content becomes too old.

Since the range of the data set period time is only one month, which is not enough to know the lifespan of the contents, the rate of change in CPV is only valid for the studied time period of the data set. Moreover, CPV is only valid at the time of measurement. Continuous logging of the user request is needed to get any new changes in the popularity of the contents and also other factors.

On the other hand, the rate of change in CPV can be used for a longer period of time like in prediction of content popularity based on the rate of user request on a video content. This can also used in adjusting cache size part assigned to that specific content.

Prefetching Factor (PF) is a parameter that elects the possible expected contents which would be the next in the watch list by each user. These contents or some parts of it should be placed in each user cache individually. These contents are predicted based on the viewing history of that user, i.e. the type or the genre of previously viewed contents by that user. The possible value of PF could be either 0 or 1 for each available content on the VOD service, i.e. 0 is assigned to the contents that should not be cached for that user, and 1 is assigned for contents that should be cached in the user cache.

The prefetching system studies the history of each user and the recently viewed contents by each user, and then decides the most probable contents that are expected to be watched by each user. These contents are placed in the users' expected watching list. This technique would effectively increase the reliability of the system and reduce network overload by placing only the needed data for each user in the cache. The expected contents which will be watched by a user could be, for example, the next episode in a series that the user had previously watched, or a similar program type the user is used to watch.

The benefit gained from the calculations of the Server Cache Setting value (SCS) and the User Cache Setting value (UCS), is to get an idea about the IPTV system needs, according to the contents' statistical properties and the user activity behavior. The result of these calculations gives the prefetching system a clue on how to designate network resources and assign some size of the cache in the server and user side to each expected contents to be viewed by each user. The prefetching system then will buffer some parts of these expected contents into the cache at the server or the user appliance.

The choice of the best place to save the cached contents is made by checking where

the buffered data will be used, i.e. if it will be used by only a few of the users or by a vast number of users. If the cached contents are only used by some of the users the prefetching system will choose to buffer these contents on each of these users' cache instead of saving it on server cache. However, if the content is popular and will be seen by a large number of users the prefetching system will buffer that content to the server cache in order to supply all of these users at the same time.

The idea of storing rarely viewed contents on user cache instead of server cache is to save space on server cache for more popular content that will be viewed by many number of users which might be an efficient way to choose the best location to save the buffered data.

Chapter 5
Result and Discussion

In this chapter, we are presenting and discussing the results of the statistical data analysis by graphs, histograms, and CDF plots. The information used in plotting these graphs in this chapter is extracted from the data set using Python. Session length and user behavior value are calculated using the equations mentioned before in Section 4.4 in Analysis Chapter.

The plotted figures in this chapter show a lot of useful information, and present many types of data analysis, which is needed for further system enhancement and service development. Lots of information can be retrieved directly from these figures which include plots, histograms, and CDF graphs of data analysis.

The resulted graphs for session length, user behavior, content popularity, and content length have been provided in each section in this chapter. In every section there is a description of each graph and also details about the relation between the result and the research questions, i.e. connection to problem formulation. Benefits or useful information that can be retrieved from these graphs are also discussed, as well as drawbacks or the information that could not be retrieved from the graphs.

Network congestion is the first problem, which affects the stability of the IPTV service, that occurs when unnecessary data are transmitted over the network. This problem can cause an overload on the server with the excessive transmitted data and can also cause network instability and degeneration. The second problem is streaming latency and start-up delay, caused by long buffering time.

Part of the solution to those problems, is understanding user behavior and session length to provide better prefetching and caching strategies. We tried to analyze these elements by studying user requests, and checking the average session length to get some values that help to decide what is the best strategy that can be implemented to enhance system performance, avoid the network congestion, and shrink the buffering time.

All of these values are determined based on the gathered data of previous user requests in the server log. The values give us the opportunity to cache the correct amount and type of content, so that it leads to decrease the transmitted data traffic, and increase the Quality of Service (QoS) as well as the Quality of Experience (QoE).

5.1 Content Length

This section studies the length of contents and describes the relationship between content length and cache arrangement. When we investigated network congestion and buffering

delay as problems, we want to solve these problems by decreasing the transmission of unnecessary data. The length of contents is one of the factors that affects the decision of cache setting adjustment by deciding how much of the video length that should be cached.

Different contents have different lengths which varies between a few minutes to several hours. The size of the content part that should be stored in the cache depends on the content length. The longer the content the longer the part that might be saved into the cache, and vice versa. The size of the content part can also be a fixed value, but it is better to make it more related to the content length. The relation between the cached part size and the content length can be measured in minutes or in percent, i.e. the part size is a portion of the original video in percent or could be some fixed values in minutes that depends on video length.

The benefit from using a variable size for the cached part of the contents is to reduce the amount of buffered data when there is a high probability that it will not be used. At the same time, the size of the cached part should be large enough to maintain smooth video streaming without time delay for buffering during the watching experience of the user.

We faced a problem in calculating the length of the contents. The problem is having duplicate contents in the data set. Many contents found in the data log that we analyzed, have one or more copies. These copies have the same EpgPID but different length, e.g., one copy is longer than the original content with a couple of minutes. The difference in length in the copy of the original content may be explained as an additional advertisement time. Other copies may include different advertisements with various lengths.

In our study, we considered the first content in the data log and discarded other copies. It is logical to take only one copy of contents with identical EpgID because we need to calculate the length of these contents only, but not multiple variations of them. This would also make the calculation of content length easier. From the data we have, there are about 46.000 contents in total. After eliminating the duplicates, there still be about 35.000 unique contents which are used to draw the two graphs in this section.

Figure 5.1 is a histogram which shows the distribution of content length, in logarithmic scale on the Y axis. The choice of logarithm scale is preferred because there are large differences between the number of contents for different tabs at assorted minutes steps in the X axis. The figure displays a broad range of content lengths which varies between 3 minutes to about 600 minutes. It is seen from the highest tab in the figure that the largest number of contents which forms about 20.000 contents have lengths with less than one hour.

In Figure 5.2, we found clearly that most users have watched contents with a length of less than one hour. The cumulative distribution function graph shows that about 85% of users have watched contents with a length of less than 100 minutes. Over 60% of the total available contents are less than one hour in length. This means that short contents are more regular than long contents. Therefore, only short parts of the contents could be cached, which are sufficient for the video streaming needs to work properly. The reduced transmitted part size helps to minimize network load, and also data loss if that part is wasted when not used if the user changes the channel or cancels the video stream.

Studying content length only to determine the requirement of the system is not enough, but in addition to content popularity which is a another important criteria that needs to be taken into account, as well as session length and user behavior that will be discussed later.

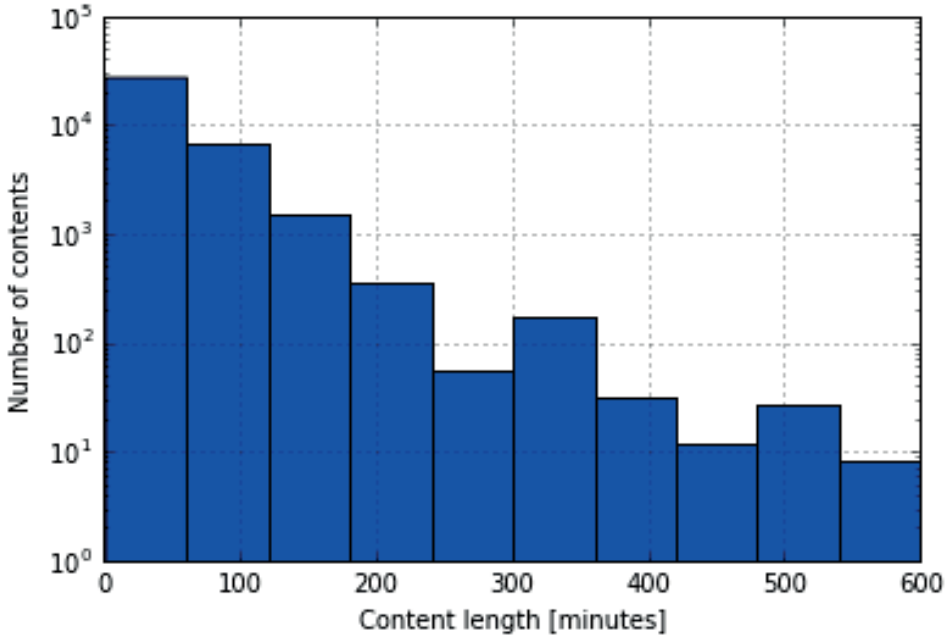

Figure 5.1: Content Length logarithmic histogram

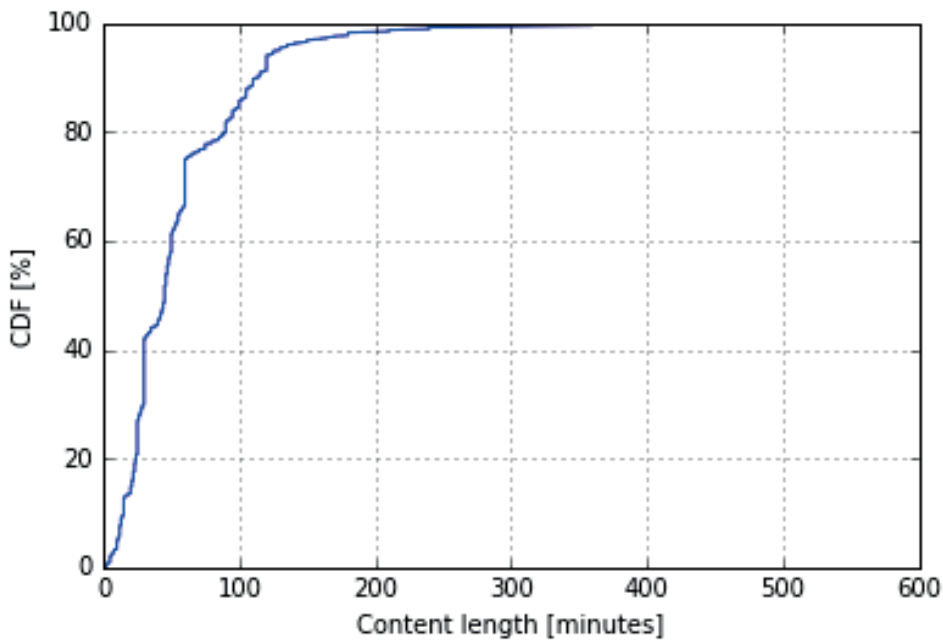

Figure 5.2: CDF of Content Length

5.2 Content Popularity

In Figure 5.3 video popularity versus number of requests is plotted to show how many user requests are registered for each video in our data log. The result is sorted in a descending order to represent the rank of each video in the list of the requested videos. It is important to know how many videos are the most popular and which types of videos the user are interested in to take more attention on high ranking videos by giving them high priority and large space in the cache.

It is shown in Figure 5.3, there is a wide difference between the highest ranked video and the lowest ranked video hence the graph is a logarithmic scale to show this large difference. Not too many videos have a very high number of views compared to the majority of videos that have only a few numbers of views. Most of the videos are requested for less than a thousand times.

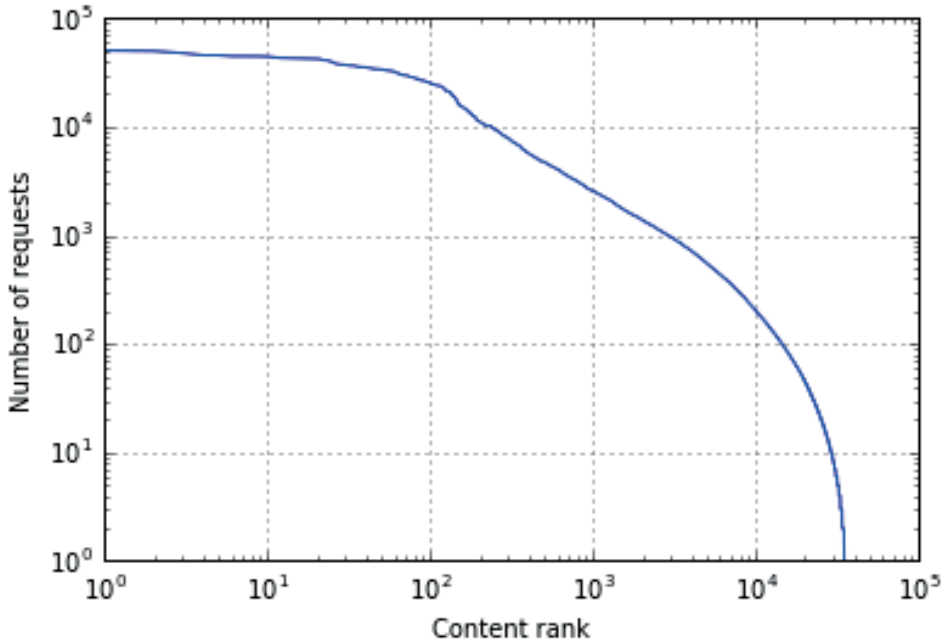

Figure 5.3: Content Popularity (log-log) plot

The benefit of Figure 5.3 is to show the number of most popular videos which in turn helps to identify those videos where large parts need to be cached to make them ready for the high demand. These popular contents are going to be cached for many users, therefore it is good to store them in the server cache to make them available for many users. The figure shows also that about 100 videos have a very high popularity with more than 20000 requests, but the popularity decreases dramatically for the rest of the videos. These high popular videos with high CPV should get high priority so that large parts of them are stored in the server cache, and they should remain a long time in the cache.

The drawback of Figure 5.3 is that it only shows the video rank in the whole period of the data set, but does not show any possible daily changes in content popularity, which

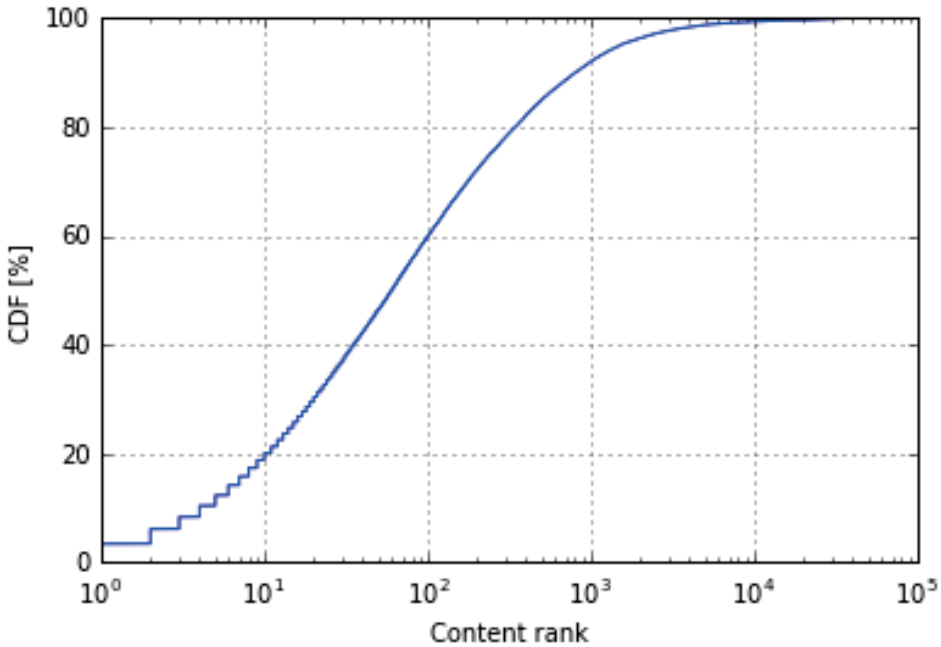

Figure 5.4: CDF of Content Popularity

may decrease or increase every day during the lifetime of the content. If we investigate the popularity of any content in the data set, we can observe that the rank of that content is not the same during the whole studied period of time. The rank of the content increases with more user requests in some days and decreases with less user requests in other days.

This drawback makes it harder to make a definite decision about content popularity that can be applied for the whole period if that content has wide changes in its popularity. However, we assume that content popularity tends to change slowly with time, and this would result in a relatively constant content popularity value CPV which would make it easier to take a definite decision about determining the needed cache size for the contents, which have relatively static ranks.

Figure 5.4 is a CDF plot, which shows how popularity is distributed on the contents. It is more visible in this figure that there are only 20% of the contents that have the highest popularity, and the rest of the contents have a much lower popularity.

5.3 Session Length of Contents

In this section, we introduce session length graphs which are plotted based on our assumptions for calculating session length by estimating the end-time for a user request from the preceding request of that user as we discussed in the previous chapter in Section 4.5.

We make the assumption that the session length is equal to the content length, when the request time is larger than the content length. However, the session length is not counted for users having only one request, and also the last request for every user is not

counted due to the lack of information about the end-time of the last session for every user. The effect of these ignored requests is small compared to the other result which is coming from the rest of the requests in the data set.

The following results and graphs are plotted and depicted according to the equation of session length in minutes and the equation of average session length in minutes, see Equation 4.1 and Equation 4.3. These two equations are described in Section 4.5.

The benefit of these graphs of session length is to get a view over how the average session length of content is distributed among the contents in order to help us to decide how many contents should be cached, how long of these video contents should be cached and how much the minimum size of cache memory is needed for these videos.

The graph in Figure 5.5 shows the average session length per content which is found after analyzing the data set and calculating the session length for every content using a simple subtraction of start-time from end-time for each content in the data set as shown in Equation 4.1. Subsequently, the average session length per content in minutes is calculated by taking the sum of the length of all sessions for each content from the whole list of requests in the data set, then dividing the result with the number of sessions for that content as shown in Equation 4.3.

The two graphs in Figure 5.5 and 5.6 are plotted for the same calculated average session length. In these two plots, the x-axis starts at 0 and ends at 450 minutes, which is the maximum measured average session length for the given data. The only difference is that the first graph has a low resolution histogram plot with a few bars, but the second graph has a high resolution histogram plot which contains 100 bars, i.e. a resolution of 4.5 minutes per bar. We used two graphs to show the same result in low and high resolution

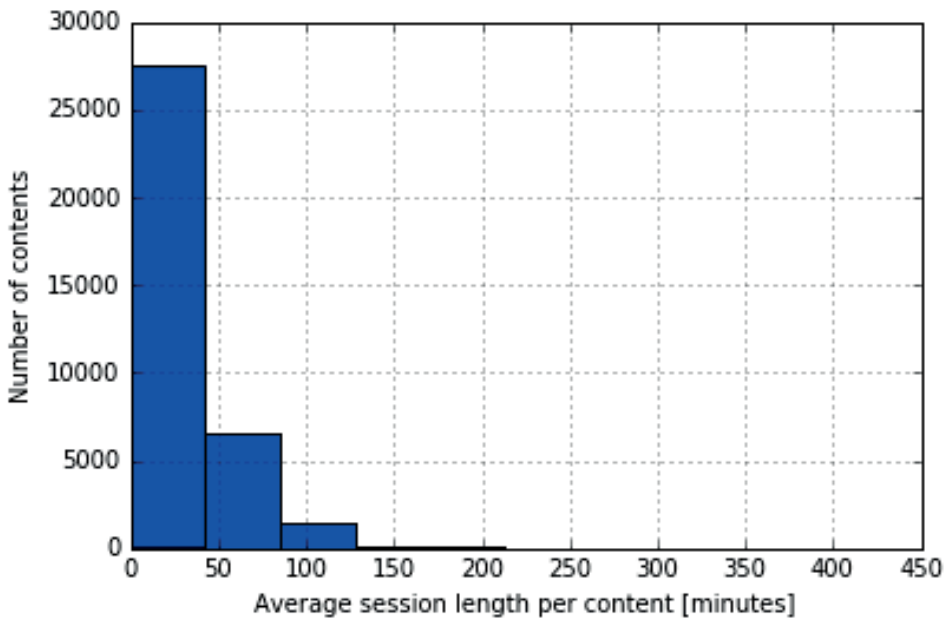

Figure 5.5: Average session length per content in minutes (low resolution)

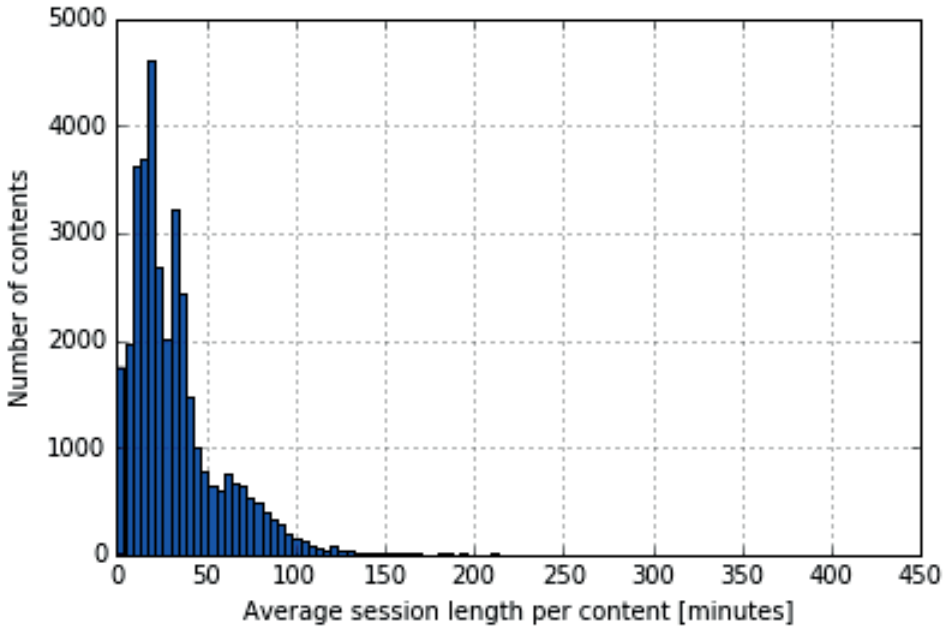

Figure 5.6: Average session length per content in minutes (high resolution)

histogram plot to see how and where the data is concentrated using wide and narrow prospects.

The same idea is also used for plotting the graph in Figure 5.7 using 10 bars, and for the graph in Figure 5.8 using 100 bars. These two graphs are plotted using the same data in the previous two graphs in Figure 5.5 and 5.6, but the only difference is that we have set a maximum limit in the x-axis in the graphs in Figure 5.7 and 5.8 to take only the first 100 minutes, i.e. x-axis range starts from 0 and limited to 100 minutes.

From the graphs in Figure 5.5 and 5.6, it is seen that most of the sessions are shorter than 100 minutes and that the sessions longer than 120 minutes are hardly visible in these two graphs. Therefore, the other two graphs in Figure 5.7 and 5.8 are plotted to display the result between 0 and 100 minutes, i.e. to show only the significant part of the first two graphs in Figure 5.5 and 5.6. On the other hand, we have to focus on short sessions, because this is the region where we can find the effect of Zappers, who watch a lot of short sessions and add much load on the server.

It is important to reduce the negative effect of Zappers by using a good caching strategy. One example of this is that would be better to cache many short sessions instead of a few long sessions in order to cover as many contents as possible ready in the cache. Only a small part of the content is supposed to be cached which will be much shorter than 100 minutes due to limited memory space and bandwidth size.

We can notice from the graphs in Figures 5.5 and 5.6 that there is a huge difference in average session length between contents. However, we know also that the content length is not the same in every program or video, and that the content length has an important effect on the length of sessions. The session length is influenced by many factors such

as content length, content popularity, and user behavior. For example, if many users are Loyal, the session length would increase, or vice versa.

Moreover, if a content is very popular its session length would also increase. However, the session length is limited by the content length so that if there are too many short videos provided, this would decrease the average session length in minutes. Nevertheless, when counting average session length in percent, the result will not show the average length of sessions for a content directly, but it will show the relation between the session length and the content length in percent.

The first thing we need to know from the graphs in Figure 5.7 and 5.8 is the peak value of the number of contents. This is a useful value that helps to adjust and choose an optimum cache size. In general, the length of the cached content part can be set based on where the majority of average sessions are located, and also the location of the highest number of contents. Thus, an optimum cached part length approaches the average session length of the highest number of contents. It can be seen in Figure 5.8 that the highest number of contents is 1200 sessions which is located at 18 minutes for the average session length per content. It is also noticed from Figure 5.7 that most of the sessions exist between 15 and 20 minutes for the average session length per content.

The maximum limit for the length of caching size per content can be set based on the length of the top number of contents and the distribution of the sessions from the graphs in Figures 5.7 and 5.8. It will be efficient to set the maximum limit of each video content to about 20 minutes and keep it in the server cache. It is not reasonable to have more than that amount of each content in the server cache. It will require a huge memory size for the cache in order to take long video contents from thousands of available video programs

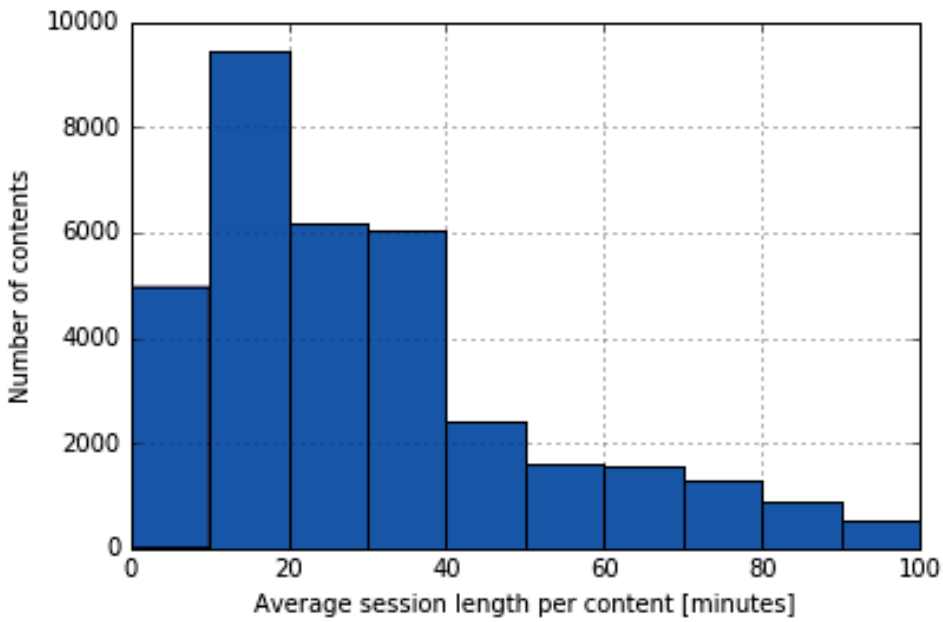

Figure 5.7: Average session length per content for (0-100) min. (low resolution)

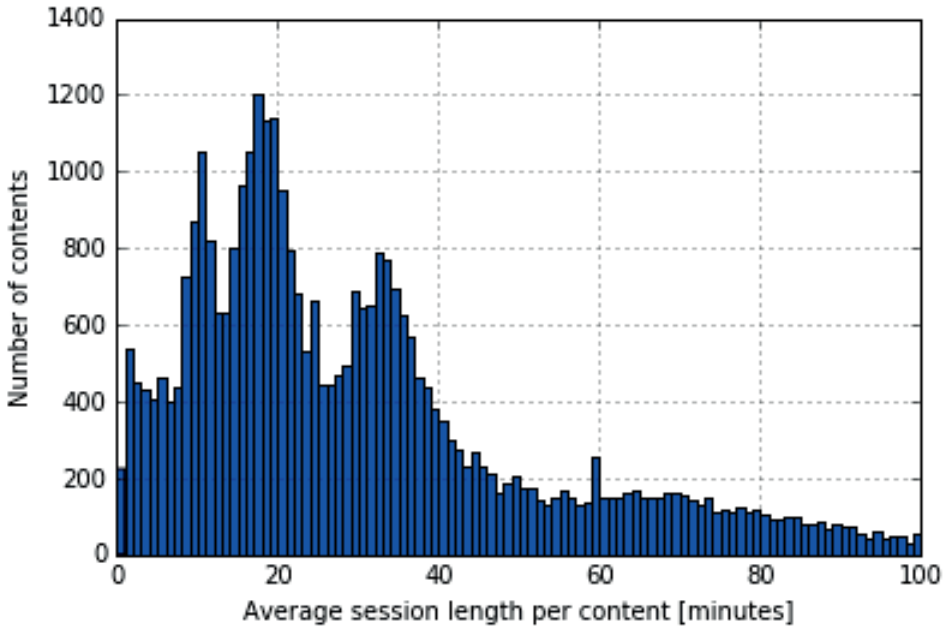

Figure 5.8: Average session length per content for (0-100) min. (high resolution)

provided by the VOD service. Therefore, it is not appropriate to have many large sized parts of many contents, but only small parts of them are feasible to be stored in the server cache.

It is noticeable that the graphs in Figure 5.7 and 5.8 have a different y-axis range because these graphs are histograms which collect and present similar and adjacent values in one bar. The number of values presented per bar depends on the total number of values and also the total number of bars used in the histogram. Therefore, the higher number of bars used in the graph and the lower number of values counted per each bar, the lower y-axis values are shown in the graph, or vice versa. Hence, the Figure 5.7 has lower y-axis values than the Figure 5.8, because it has more bars with a low number of contents per bar, i.e. number of contents that is counted and included in each bar is low, this appears in low bars in graph 5.8. A total of 100 bars are used in graph 5.8, where each bar represents one minute of average session length per content. However, only 10 bars are used in graph 5.7, where each bar represents ten minutes of average session length per content.

We also plotted CDF graphs 5.9 and 5.10 to help us understand histograms from another point of view and to show the distribution of sessions using cumulative distribution function plots. It is easier to read the distribution of session length and to know the length of the majority of sessions. The first CDF graph is based on all requests in the data set. It can be seen that almost 90% of the average session lengths are under 60 minutes. It is more clear to check this ratio in the second CDF graph 5.10 which displays the distribution of average session lengths that are up to 100 minutes. By checking the ratio at 20 minutes, about 40% of average sessions are less or equal to 20 minutes. This value is significantly large compared to the longer average sessions per content.

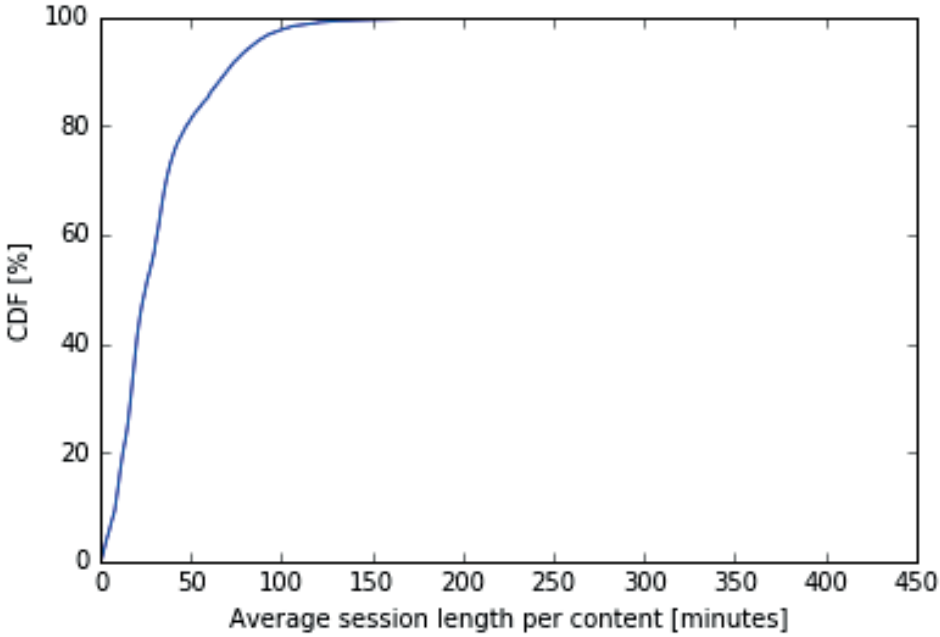

Figure 5.9: CDF of Average session length per content for (0-450) min.

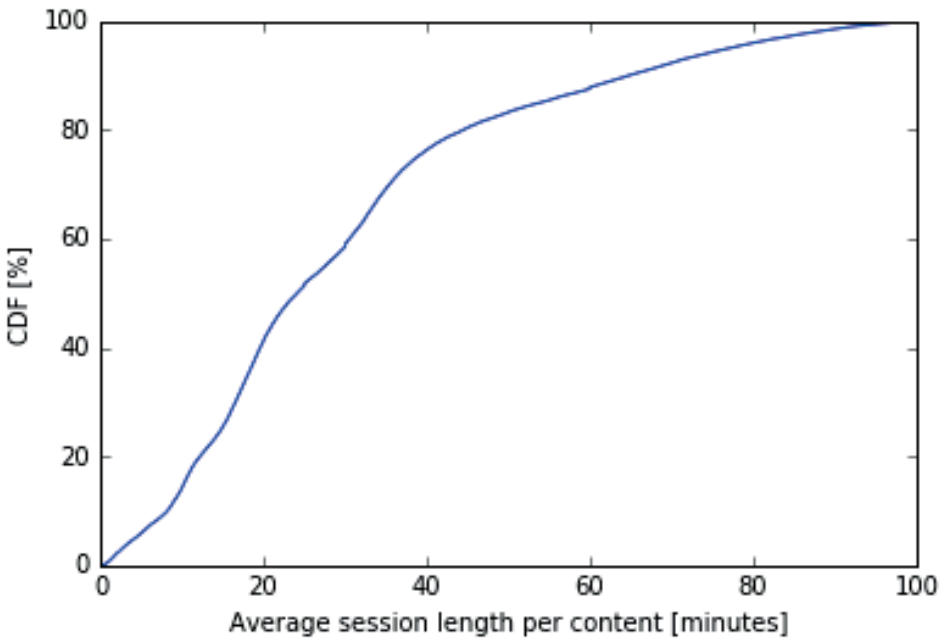

Figure 5.10: CDF of Average session length per content for (0-100) min.

One drawback in plotting the average session lengths is that all these graphs are representing the average value of sessions per content and not the actual length of sessions. The actual sessions might look different if plotted directly in the graphs without taking the average. The sessions related to all users and contents will be scattered over the x-axis and interfere with each other, as well as the top number of similar or adjacent sessions may be located differently compared to the calculated average. However, it is hard to plot the actual sessions directly in a graph because there are over 16 millions sessions, which means that it needs a lot of computer memory and processing power to plot them using a personal computer. Thus, Python program kernel stopped working when we tried to process and draw that huge amount of data. However, calculating and plotting the average value is more efficient and enough to represent the needed results from the data set.

All the previous results and graphs were measured in minutes, but the following two graphs will display the result of calculating average session length per content in percentage value instead of minutes. The graphs are drawn according to the results obtained by calculating and analyzing the data using Equation 4.2 to get session length in percent and using Equation 4.4 to get the average session length in percent.

Figure 5.11 shows the average session length per content in percent. It is visible that there is a big difference in the shape of the graph in Figure 5.5 which uses [minutes] as a unit of measurement and the graph in Figure 5.11 that uses [percent] as a unit of measurement. The main difference is that when calculating real time in minutes, the result will also be in real time unit and gives directly related results that reflect the real values of session length. But when using percent, the result of each session length calculation will be divided by that session length which gives comparative results.

The majority of video contents in Figure 5.11 have an average session length between

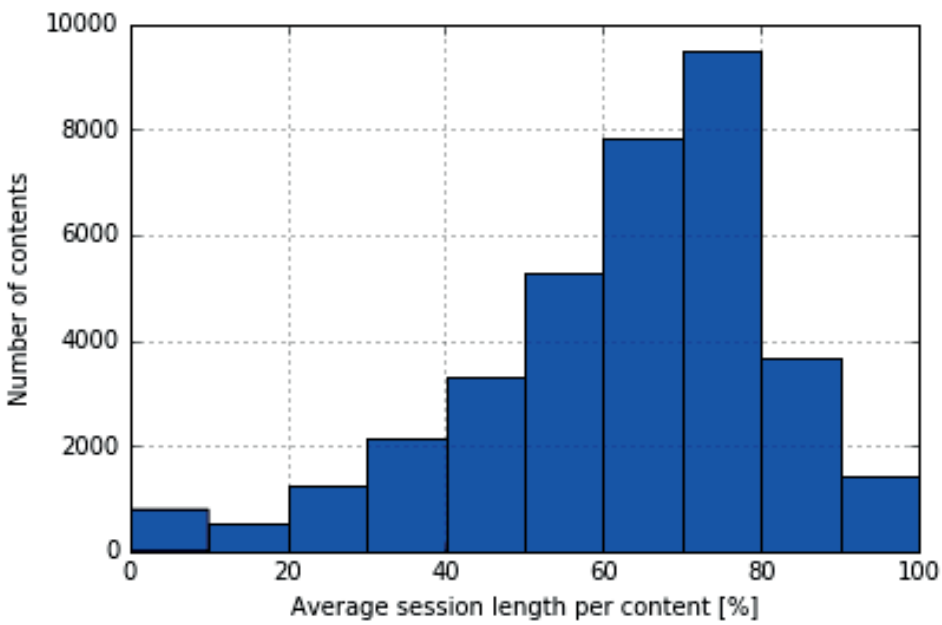

Figure 5.11: Average session length per content in percent

40% and 90%. The peak value for the number of contents is over 9.500 videos, which have an average session length of 75%. By dividing the peak value for the number of contents with the total number of contents which is approximately 35.000 videos, the peak value accounts for about 27% of the total available contents.

The advantage of plotting the average session length in percent is that it gives an overview of how much of the content have been viewed for each content on average. This would result in deciding how much of each content should be cached individually based on the calculated percentage average session length for that content.

The disadvantage of calculating the average session length in percent per content is that it ignores user behavior, but focuses on the content only regardless of which user has requested that content. This gives more benefit for adjusting the server cache than the user cache because in general, since the server cache is for all users but the user cache is only for that specific user. However, this calculation can be used to help in determining cache size for each content in server cache and also in user cache.

A CDF plot of average session length per content in percent is added to check the distribution of sessions per content in percent and to calculate how many contents have been seen up to a specific viewing ratio of average session length over content length.

In Figure 5.12, the CDF of the average session length per content in percent is shown. We can notice from the CDF plot that about 60% of the total available video contents are viewed with an average session length of up to 75% of its content length. Only 20% of total video contents are viewed with a rate of up to 50%, i.e. the average session length is equal or less than half of the video content length.

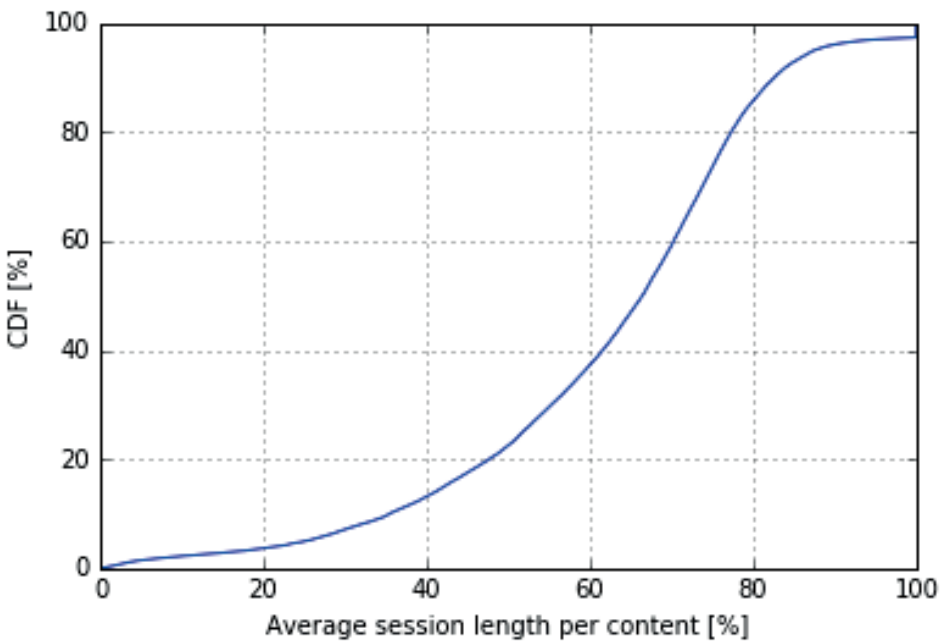

Figure 5.12: CDF of Average session length per content in percent

5.4 Loyal vs Zapper behavior

In this section, the result of user behavior is demonstrated. The average session length per user in percent is plotted in Figure 5.13, which shows the user behavior for all users. An important result from the study of user behavior is to calculate the User Behavior Value (UBV), which determines the loyalty of a user. This calculation takes into account the session length calculation assumption we adopted earlier in the same way used in the calculation of contents session length in Section 5.3. The only difference here is that the average of sessions length is calculated by taking the sum of all sessions viewed by a user and dividing the sum with the number of sessions viewed by that user, i.e. the average is per user not per content. This calculation is shown in Equations 4.5 and 4.6, which are described in Section 4.6.

The negative value in Figure 5.13 illustrates the number of non-active users who are excluded from our calculations. There are about 40.000 users exempted from the user behavior calculations because they have only one request during the logging time, in other words, missing the end-time of their session. Non-active users comprise about 8% of the total number of users. However, it would add a significant increment on 100% Loyal users if we consider these non-active users as 100% Loyal users by assuming that they have watched the only one content they requested in the studied month of the data set.

The values of the average session length per user symbolized as UBV, is between 0 and 100%, or in another description 0 and 1. When UBV for a user approaches 1, this means that this user approaches 100% loyalty behavior. This can be explained by the fact that this user nearly always watches the full-length of any requested content. However, if

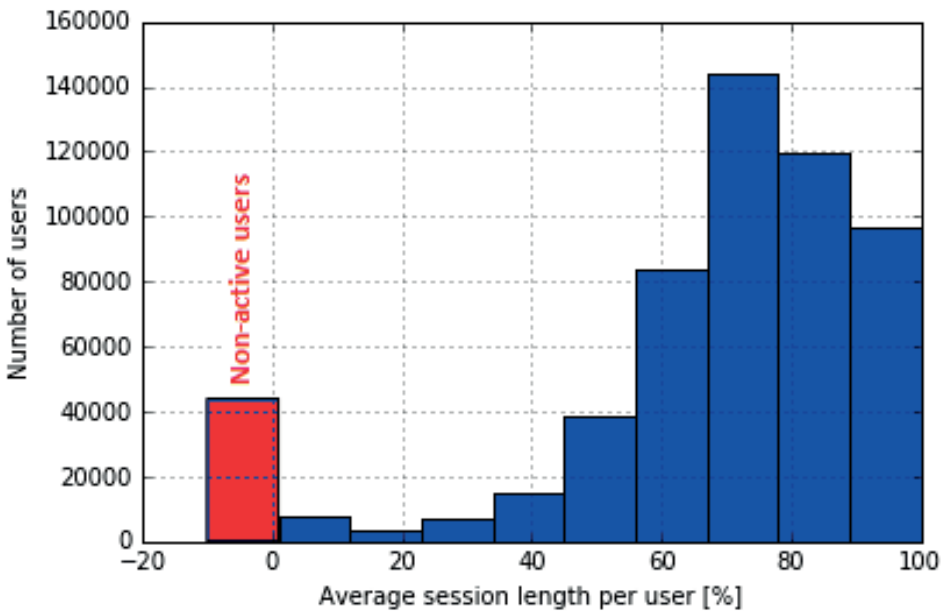

Figure 5.13: Average session length per user in percent

UBV for a user approaches 0, this means that this user approaches 100% Zapper behavior. This can be explained by the fact that this user tends to not complete watching any content and keeps browsing between various contents.

The advantage of the Figure 5.13 is that it provides an overview of the behavior of users and shows the loyalty percentage level of each user that is evaluated based on its calculated User Behavior Value (UBV). This value can help the prefetching engine to decide a better caching strategy into the server and user cache according to user loyalty. Loyal users can get larger parts of the videos ready in the cache. Zapper users can get smaller parts of the videos in order to reduce the load on the server and the network, and to minimize the wasted bandwidth on unused large transmitted video parts.

The drawback of Figure 5.13 is that the shape of the plot varies depending on the assumed ratio, which is discussed in Session Length Calculation Assumptions Section 4.5 in Page 21. This ratio is treated throughout our calculations as 100%, whenever a user makes a request after the end of the content length. The resulted shape of the plot might not represent the actual values of the sessions since these values are calculated based on our assumption which has a big influence on average session length values, that it might not mirror the real case of the actual session length. The loyalty behavior is affected by some assumptions we have made that might not give the exact real values, but it is within a range of possible values. This range relies directly on the assumed ratio which equals to 100%. This ratio is used in all calculations for any session when the next content request comes after the end of the current requested content of that session. If we choose another value for the assumed ratio, the result of the user behavior will change and bias towards that new value.

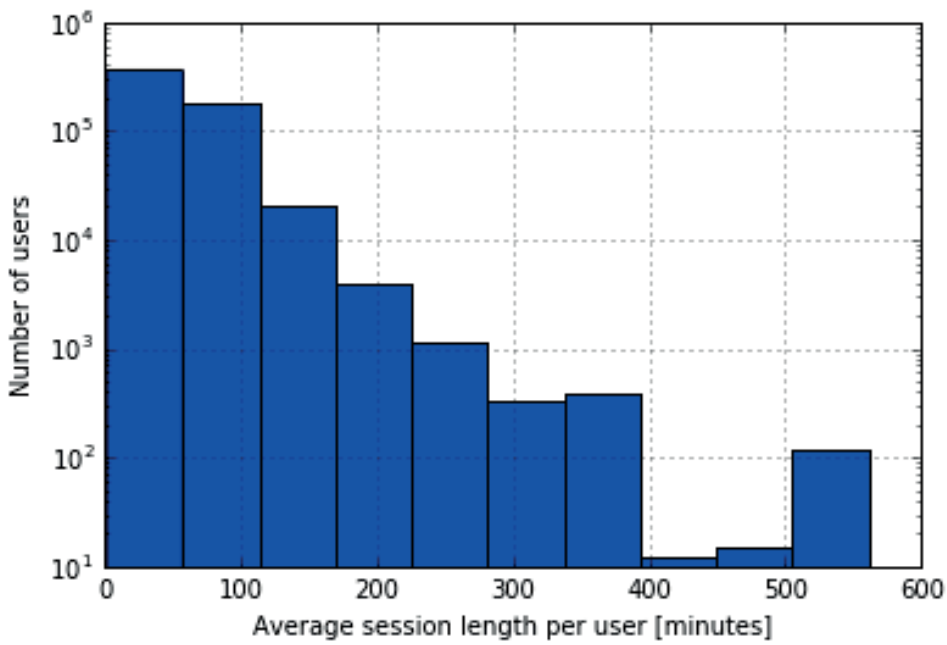

Figure 5.14: Average session length per user in minutes

Figure 5.14 represents the average session length per user in minutes. The graph in Figure 5.14 shows absolute values in minutes, whereas the graph in Figure 5.13 shows relative values in percent. Moreover, the average session length per user in minutes does not represent the loyalty of the users, since the user loyalty depends on how long the user have seen from the contents, and these contents have different lengths. We think that the relative values in Figure 5.13 can show the loyalty of the user better than the absolute values. However, we can still find the Zappers, as we did using Figure 5.13, if we plot a graph that shows the average session length per user in minutes from 0 to 30 minutes, for example, if we consider that the maximum value for average session length for Zappers is 30 minutes.

The peak value, which can be seen in the first bar to the left in Figure 5.14, means that about 350.000 users have seen less than 60 minutes for the average session length. For the next bar, there are about 180.000 users have seen between 60 and 120 minutes. For the rest of bars, number of users decreases dramatically after 120 minutes to less than 20.0000.

Figure 5.15 shows the CDF plot of user behavior in percent. The x-axis represents the average session length per user in percent, and the y-axis represents the complementary distribution function (CDF). We can read from Figure 5.15 that almost 40% of users have a user behavior value less than 70%, in other words, about 60% of the total users have UBV over 70%. Users with high UBV, i.e. over 50%, are regarded as Loyals, which comprise nearly 90% of the total users. On the other hand, users with low UBV, i.e. under 50%, are regarded as Zappers, which account for approximately 10% of the total users.

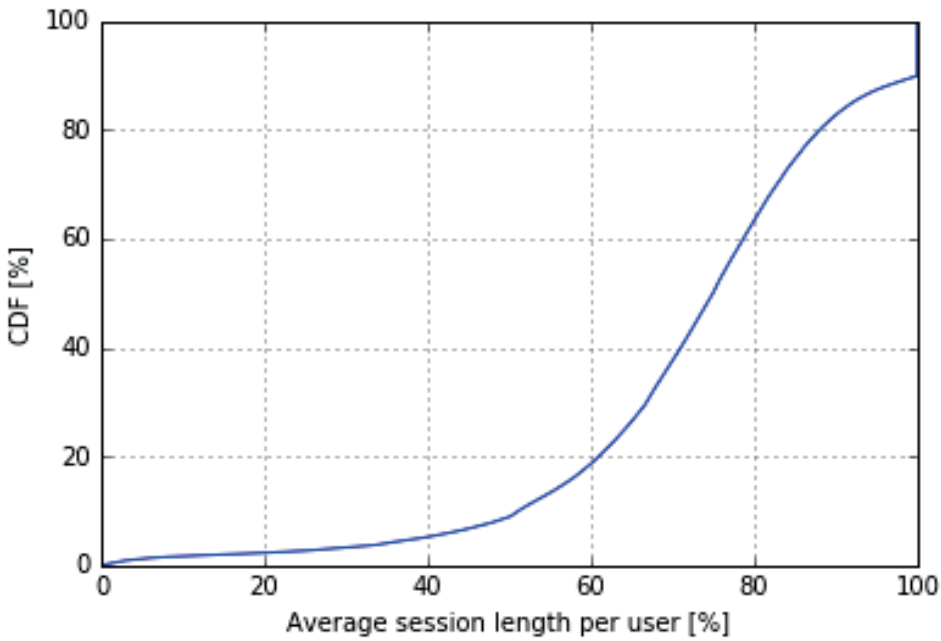

Figure 5.15: CDF of User Behavior in percent

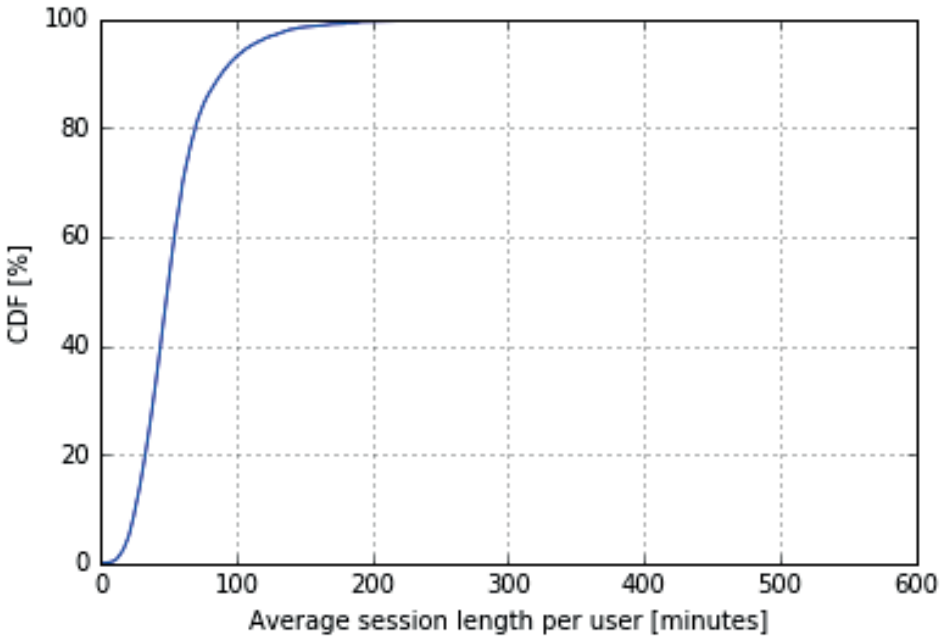

Figure 5.16: CDF of User Behavior in minutes

Figure 5.16 demonstrates the CDF plot of user behavior in minutes. The x-axis denotes the average session length per user in minutes, and the y-axis denotes the complementary distribution function (CDF). We can notice from Figure 5.16 that it has a very sharp slope between 0 and 100 minutes with a CDF between 0 and about 95%, which means that most of the sessions are in that region. We can see that 90% of the average session length per user is shorter than 100 minutes. This means that very few users have a very long session length. Therefore, we have to pay more attention to short sessions by providing the required resources to take into account 100 minutes as the longest session length for the majority of the users.

Chapter 6
Conclusion

The result of this thesis work can be concluded in brief points which include the summary of the main research subjects. Those subjects identify the basis on which the work is done and cover user behavior, content popularity, content length, and session length. The relationship between all of these subjects and the prefetching and caching system, have also been investigated to get a clear image about its influence in the system and to extract variables which can be used to formulate equations to calculate the best caching strategy and manage the prefetching system. Those results from the data analysis can be implemented to develop the current methods used in prefetching and caching systems in order to improve system and network performance.

The limitation in the possibility of calculating correct session length is solved by taking assumptions. Average session length can describe either user behavior, or content usage, if it is calculated per user or per content respectively. Two values extracted from the calculations of average session length, are User Behavior Value (UBV), and Content Session Value (CSV), which are related to the users and the contents respectively. The higher these values are, the larger part of the content should be devoted to the respective users or contents. Since UBV is linked to each user, i.e. each user has its own value, the outcome of the cache and prefetching modification is concerned to each user separately and also affects each user cache separately. On the other hand, CSV is linked to each content which can be viewed by any user. Therefore, the outcome of the calculation related to the contents will affect many users, thus it should be implemented in the server cache.

The objective of the session length calculations was to explore how much the users used to see on average, and how long part of the content is expected to be seen by users. It also helps to decide the proper length of each content part, which should be cached in the user cache or the server cache. Thus, preserving a high QoS and QoE, as well as a low network load.

The study of content length presented the contribution of all data and provided a good picture about the lengths of contents. The reason to study content length is to make a decision about how much is enough to take from the contents in the cache based on the distribution of the length of all available contents. The size of the cached part should be flexible instead of a fixed value. This value changes relatively with the available content length. The calculation of the size of the cached part can be done either in percent, or in minutes. Finally, one of the two options is chosen after comparing the outcomes for each one, then the preferred option is selected, or both options are implemented cooperatively.

The goal of studying content popularity is to find a solution for reducing unnecessary data traffic by identifying popular videos and preparing the system to give popular videos higher priority. This is done by reserving more resources to those videos, i.e. by designating more cache size to them, and storing them in the cache for a longer time than the other videos.

User Behavior is a crucial factor when calculating the correct amount of data that should be cached. Section 5.4 Loyal vs Zapper behavior discussed the UBV and the effect of this value to decide a better caching strategy into the user cache according to user loyalty. The cached data size depends on the user behavior value, thus it increases if the user is Loyal, and decreases if the user is Zapper. This leads to a reduction of the transmission of unnecessary data. Consequently, network congestion can be avoided.

As a summary, we can conclude the following possible caching strategies for both the server side and the user side. The scheme of these strategies can be found in the flow chart we have drawn previously, see Figure 3.1. It is important to note that none of the terms alone can give full benefit to the system, and they have to work together to reach the best cache setting. An adaptive cache settings based on the activity of the users and the properties of the contents should be implemented to increase the possibility to set the correct settings for the prefetching system.

- Server cache: The server makes a decision about which video is going to be prefetched and how much is the length of content part that will be cached. For example, the server can cache 10 minutes or less from all of popular videos, which in our case, were less than 1000 videos, see Figure 5.4. That arbitrary number of cached minutes in the example actually depends on the result of the data analysis.
- User cache: When a user is identified as a Loyal by having a habit of seeing long portion of the videos i.e. the user has a high user behavior value, the system starts to cache longer parts of the prefetched contents. However, this gives an impact of caching fewer number of contents due to the increasing large sum of the cached data size and the limited free space of the cache. In the other case when Zappers tend to view only short parts of many contents, the system will act by caching small parts of those contents in order to save space on the user cache memory and to keep providing those users with the requested content in a speedy way.

Chapter 7
Future Work

There are a few things that could be added to this project to improve it, but were not added due to the lack of space and time. The implemented data set in this thesis is a log of user requests over one month. That period of time is taken as one piece during the analysis of the data set. Details that need further study are analyzing daily user activity which includes user habits during the day and the evening time, as well as hourly user requests. Diagrams related to these analyses also need to be plotted.

A plot for the most popular content or the first rank video could be used as an example to show the change in popularity with time along the whole period of the data set log time. This could be a helpful way to make an overview about dynamic content popularity and to know if it is growing or shrinking with time. That will add more precise values to our calculation of content popularity and give more flexibility to the system.

In the calculation of average session length we made some assumptions about the possible session length end-time, but we did not use all of them and we only implemented full-time session in the graphs and calculations when the information about session end is missing. Other assumptions may also be implemented and used in the calculation of session length, and in the plots. For example, drawing other diagrams to show average session length with a new assumed ratio for session length, such as half of the session length or 0% of it, that could be used to study new possible user behavior and session length values.

Information about the network speed at the end terminal, is an unaddressed factor that could be utilized in taking the decision of the prefetching and caching processes. This factor could be exploited by the server to have an adaptive cache setting based on the connected network speed to the end terminal. Thus, it would save system and network resources by maintaining an adequate resources to each user based on the maximum connected speed of the end terminal.

Network connection could be used as the top limit factor of the prefetching system, while the other factors discussed in this project can be used along with that factor. It is worthless if the server spends large bandwidth and cache resources to a user with low connection speed, who will be affected with low QoE. However, the user streaming quality can be boosted by preloading the requested video in good time to get more smoothness in watching the requested content. Thus, QoE for users with low connection speed could be enhanced.

For users with high connection speed at the end terminal, there is no need for a large user cache. Hence, the connectivity of these users is sufficient to view the requested

content, even in high definition quality, using small user cache size. An example of the previous description is the difference between the first version and the second version of Google Chromecast media streaming device.

Both versions of Google Chromecast are used mainly for watching online video contents like YouTube videos on a TV by using, for example, mobile devices as a remote control. However, it is also possible to view local video casts from mobile devices on a TV in the same local network. Although, the second version of the Google Chromecast device comes with lower cache size than the first version, but according to Google Inc., the newer version gives 2 to 4 times speed improvement compared to the first version. This is not only referred to the improvement in the processor speed of the new device, but also to the improvement in network connectivity which is based on 802.11ac vs. 802.11n in the previous version of that device [31].

As a result the higher the connected network speed of the terminal is, the lower user cache size is needed, because the speed of the network is sufficient to deliver many parts of the requested content fast enough to be viewed by the user, before the user cache get filled with large parts of the cached content data.

It would be better to add that factor by distributing server resources to all users based on the connectivity speed of each end terminal. However, in the future, it might not be a large problem because the network connection speed is always getting higher and higher. Therefore, it is not crucial to have this factor, but it will always give an advantage if we take in mind that there could be some network congestion in any network due to the huge traffic and the mass user demand.

References

[1] 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*. IEEE. 2009, pp. 1–6.

[2] M. L. Glaser. *IPTV PRIMER: What is IPTV and is it Regulated as a Traditional Cable Service?* URL: http://www.telecomattorneys.com/iptv-primer.html (Accessed: May 2017).

[3] N. Narang. *#2 Concept Series : What is the Difference between OTT and IPTV*. Apr. 2013. URL: http://www.mediaentertainmentinfo.com/2013/04/2-concept-series-what-is-the-difference-between-ott-and-iptv.html/.

[4] M. Rouse. *Definition IPTV (Internet Protocol television)*. URL: http://searchtelecom.techtarget.com/definition/IPTV (Accessed: Mar. 2017).

[5] *Internet Protocol Television (IPTV)*. URL: http://www.techopedia.com/definition/24957/internet-protocol-television-iptv (Accessed: Jan. 2017).

[6] C. Woodford. *IPTV*. (Last updated: Jan. 23th, 2017). URL: http://www.explainthatstuff.com/how-iptv-works.html.

[7] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper*. Feb. 2017. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[8] J. Li, A. Aurelius, V. Nordell, M. Du, Å. Arvidsson, and M. Kihl. “A five year perspective of traffic pattern evolution in a residential broadband access network”. In: *Future Network & Mobile Summit (FutureNetw), 2012*. IEEE. 2012, pp. 1–9.

[9] *YouTube Company Statistics*. (Last updated: Sep. 1st, 2016). Statistic Brain. URL: http://www.statisticbrain.com/youtube-statistics/.

[10] E. Harrie. *Increasing video-on-demand viewing in the Nordic countries*. Apr. 2016. URL: http://www.nordicom.gu.se/en/media-trends/news/increasing-video-demand-viewing-nordic-countries.

[11] J. O'Neill. *Global SVOD growing faster than thought; OTT revenues could reach $51B by 2020*. June 2015. URL: http://www.ooyala.com/videomind/blog/global-svod-growing-faster-thought-ott-revenues-could.

[12] *Movies the Main Draw for UK Video-on-Demand Subscribers*. Aug. 2014. URL: http://www.emarketer.com/Article/Movies-Main-Draw-UK-Video-on-Demand-Subscribers/1011130.

[13] C. Grece, A. Lange, A. Schneeberger, and S. Valais. "The development of the European market for on-demand audiovisual services". In: *European Audiovisual Observatory* 174 (2015).

[14] *CNC Dossiers no. 318 May 2011, / Video on Demand and Catch-up TV*. URL: http://www.cnc.fr/web/en/index?p_p_auth=7LlBvPYK&p_p_id=20&p_p_lifecycle=1&p_p_state=exclusive&p_p_mode=view&_20_struts_action=%2Fdocument_library%2Fget_file&_20_folderId=136213&_20_name=DLFE-3183.pdf.

[15] *CNC Dossiers no. 322 May 2012, / Video on Demand and Catch-up TV*. URL: http://www.cnc.fr/web/en/index?p_p_auth=7LlBvPYK&p_p_id=20&p_p_lifecycle=1&p_p_state=exclusive&p_p_mode=view&_20_struts_action=%2Fdocument_library%2Fget_file&_20_folderId=16538&_20_name=DLFE-4630.pdf.

[16] S. M. Thampi. "A Review on P2P Video Streaming". In: *arXiv preprint arXiv:1304.1235* (2013).

[17] M. Du, M. Kihl, Å. Arvidsson, H. Zhang, C. Lagerstedt, and A. Gawler. "Prefetching schemes and performance analysis for TV on demand services". In: *International Journal on Advances in Telecommunications* 8.3&4 (2015), pp. 162–172.

[18] S. Gawade and H. Gupta. "Review of Algorithms for Web Pre-fetching and Caching". In: *International Journal of Advanced Research in Computer and Communication Engineering* 1.2 (2012), pp. 62–65.

[19] A. Balamash, M. Krunz, and P. Nain. "Performance analysis of a client-side caching/prefetching system for Web traffic". In: *Computer Networks: The International Journal of Computer and Telecommunications Networking* 51.13 (2007), pp. 3673–3692. DOI: 10.1016/j.comnet.2007.03.004.

[20] A. Silberschatz, J. Peterson, and P. Galvin. *Operating system concepts*. Addison Wesley, Reading, 1992, pp. 334–339.

[21] S. K. Das, Z. Naor, and M. Raj. "Popularity-based caching for IPTV services over P2P networks". In: *Peer-to-Peer Networking and Applications* 10.1 (2017), pp. 156–169. DOI: 10.1007/s12083-015-0414-3.

[22] S. Chen, B. Shen, Y. Yan, and X. Zhang. "Buffer Sharing for Proxy Caching of Streaming Sessions". In: (2003). URL: http://pdfs.semanticscholar.org/ddcf/68b6dd3b22684e46514acd5d173d8788eab7.pdf.

[23] A. Ali Eldin, M. Kihl, J. Tordsson, and E. Elmroth. "Analysis and Characterization of a Video-on-Demand Service Workload". In: (2015), pp. 189–200. DOI: 10.1145/2713168.2713183.

[24] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. "Understanding user behavior in large-scale video-on-demand systems". In: *ACM SIGOPS Operating Systems Review*. Vol. 40. 4. ACM. EuroSys, 2006, pp. 333–344.

[25] L. Guo, S. Chen, Z. Xiao, and X. Zhang. "Analysis of Multimedia Workloads with Implications for Internet Streaming". In: (2005), pp. 519–528. DOI: 10.1145/1060745.1060821.

[26] The Pyzo team. *Python vs Matlab*. URL: http://www.pyzo.org/python_vs_matlab.html (Accessed: Apr. 2017).

[27] B. Klein. *Python Course, Numerical Python, Numpy Tutorial*. URL: http://www.python-course.eu/numpy.php (Accessed: Mar. 2017).

[28] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[29] Wes McKinney & PyData Development Team. *Pandas: Powerful Python data analysis toolkit*. URL: http://pandas.pydata.org/pandas-docs/stable/ (Accessed: May 2017).

[30] A. Sharma, A. Goel, et al. "A Framework for Prefetching Relevant Web Pages using Predictive Prefetching Engine (PPE)". In: *arXiv preprint arXiv:1109.6206* (2011).

[31] M. Grothaus. *Chromecast 2 vs. Chromecast: The Best Just Got EVEN BETTER*. URL: http://www.knowyourmobile.com/devices/google-chromecast-2/23327/chromecast-2-vs-chromecast-best-just-got-even-better-comparison (Accessed: Apr. 2017).

Printed by Tryckeriet i E-huset, Lund 2017

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2017-594

http://www.eit.lth.se