Master Thesis Report:
Analyzing YouTube Content Demand Patterns and Cacheability in a Swedish Municipal Network

Master of Science Thesis

Hantao Wang
hantao@kth.se

Supervisor: Jie Li (jie.li@acreo.se)
Acreo AB, Swedish ICT, Sweden

Examiner: Prof. Björn Knutsson (bkn@kth.se)
KTH Royal Institute of Technology, Sweden
School of Information and Communication Technology (ICT)
KTH Royal Institute of Technology
Stockholm, Sweden
November 15, 2012
Abstract

User Generated Content (UGC) has boosted a high popularity since the birth of a wide range of web services allowing the distribution of such user-produced media content, whose patterns vary from textual information, photo galleries to videos onsite. The boom of Internet of Things and the newly released HTML5 accelerate the development of multimedia patterns as well as the technology of distributing it. YouTube, as one of the most popular video sharing site, enjoys the topmost numbers of video views and video uploads per day in the world. With the rapid growing of multimedia patterns as well as huge bandwidth demand from subscribers, the sheer volume of the traffic is going to severely strain the network resources.

Therefore, analyzing media streaming traffic patterns and cacheability in live IP-access networks today leads a hot issue among network operators and content providers. One possible solution could be caching popular contents with a high replay rate in a proxy server on LAN border or in users’ terminals.

Based on the solution, this thesis project focuses on developing a measurement framework to associate network cacheability with video category and video duration under a typical Swedish municipal network. Experiments of focused parameters are performed to investigate potential user behavior rules. From the analysis of the results, Music traffic gets a rather ideal network gain as well as a remarkable terminal gain, indicating that it is more efficient to be stored close to end user. Film&Animation traffic, however, is preferable to be cached in the network due to its high net gain. Besides, it is optimal to cache the video clips with a length between 3 and 5 minutes, especially the Music and Film&Animation traffic. In addition, more than half of the replays occur during 16.00-24.00 and peak hours appear on average from 18.00 to 22.00. Lastly, only around 16% of the videos are global popular and very few heavy users tend to be local popular video viewers, depicting local limits and independent user interests.

Key words: UGC, QoE, YouTube, cacheability, network gain, data mining, massive data analysis, experiment framework, video category, video duration
Sammanfattning
Acknowledgements

First of all, I would like to give my acknowledgement to the best examiner Prof. Björn Knutsson for guiding me and inspiring me all the way around during this thesis project period. I could not be more grateful for his detailed and elaborate feedbacks which help me self-improving and growing with this work.

Secondly I would like to express my sincere gratitude to my industrial supervisor Jie Li for his kind guidance from the very beginning till the end of the thesis work. He is a responsible advisor as well as an intimate mentor during the past half year. I enjoyed every single discussion with him and appreciated for his valuable advices and constant support throughout the whole project.

In addition, I must show my special appreciation to Prof. Åke Arvidsson from Ericsson who showed continuous interest in this thesis work. He has always been willingly to share his wide knowledge and has contributed a lot to this work by having regular discussions with me and providing me with his innovative ideas.

I also want to thank Andreas Aurelius, Viktor Nordell, Manxing Du, Belgis Chial and all the people from Acreo who have helped me out during this abundant journey.

Finally, I would like to my parents, who always support me and encourage me to get through the ebb of my life.

Hantao Wang

Stockholm, 15th Nov 2012
# Table of Contents

Abstract........................................................................................................................................... i

Sammanfattning ................................................................................................................................. ii

Acknowledgements ............................................................................................................................ iii

Table of Contents ............................................................................................................................... iv

Chapter 1  Introduction ..................................................................................................................... 1

  1.1 Overview ..................................................................................................................................... 1

  1.2 Goals .......................................................................................................................................... 5

  1.3 Project Scope ............................................................................................................................. 6

  1.4 Expected Outputs ....................................................................................................................... 7

  1.5 Report Composition .................................................................................................................. 8

Chapter 2  Backgrounds .................................................................................................................... 9

  2.1 User Behaviors and QoE .......................................................................................................... 9

  2.2 Locality-Aware Networks and Cacheability .......................................................................... 10

  2.3 YouTube ................................................................................................................................... 11

  2.4 Previous Analysis on YouTube Traffic ................................................................................... 14

Chapter 3  Platform and Methodology ............................................................................................. 17

  3.1 Main Approach ....................................................................................................................... 17

  3.2 Network Architecture ............................................................................................................. 18

  3.3 Aggregation Network Patterns ............................................................................................ 19

  3.4 Data Collection Framework ................................................................................................. 20

  3.5 Data Warehouse ..................................................................................................................... 23

  3.6 Data Analysis Method ............................................................................................................ 26

Chapter 4  Experiments and Analysis ............................................................................................. 29

  4.1 Basic YouTube Statistics Characteristics ............................................................................. 29

  4.2 Replays and Simultaneous Downloads ............................................................................... 36

  4.3 Network Gain/Terminal Gain with Category ....................................................................... 38
4.4 Network Gain with Video Duration .................................................. 47
4.5 Problem and Hypothesis ...................................................................... 49
4.6 Summary ............................................................................................... 51
Chapter 5 Conclusions ............................................................................. 53
  5.1 Conclusions from Backgrounds ............................................................ 53
  5.2 Conclusions from Goals ....................................................................... 53
  5.3 Conclusions from Methodology ............................................................ 53
  5.4 Conclusions from Experiment and Analysis ........................................ 54
  5.5 Personal contributions ........................................................................ 56
Chapter 6 Future Work .............................................................................. 57
References ................................................................................................. 58
List of Figures

Figure 1-1 Video traffic explosion of global traffic and mobile traffic ........................................ 2
Figure 1-2 Video Revenue of Four Continents ................................................................. 3
Figure 1-3 CDN network architecture .................................................................................. 4
Figure 1-4 Bandwidth consumption statistics in Europe ....................................................... 5
Figure 1-5 Spotify traffic type proportion ............................................................................... 5
Figure 2-1 Share of videos in popular sites ............................................................................. 12
Figure 2-2 Basic YouTube video retrieval .............................................................................. 13
Figure 2-3 Data Parsing from YouTube data API ................................................................. 14
Figure 3-1 General Approach of the Project ......................................................................... 18
Figure 3-2 Overall Network Architecture ............................................................................ 18
Figure 3-3 Municipal network architecture ......................................................................... 19
Figure 3-4 Data Collection Process ...................................................................................... 20
Figure 3-5 Pre-processing with Python Scripts ................................................................. 23
Figure 4-1 User type distributions ....................................................................................... 30
Figure 4-2 Proportions of YouTube video categories ......................................................... 30
Figure 4-3 Histogram of Video Duration ............................................................................. 31
Figure 4-4 Video duration in regard to categories .............................................................. 31
Figure 4-7 View counts ........................................................................................................ 32
Figure 4-8 Favorite counts ................................................................................................... 33
Figure 4-9 Rating .................................................................................................................. 33
Figure 4-10 Fractions of global popular requests and videos ............................................... 34
Figure 4-11 Traffic of local popular videos ......................................................................... 35
Figure 4-12 Time trends of video traffic in various categories .............................................. 35
Figure 4-13 Date distribution of traffic for categories ......................................................... 36
Figure 4-14 Time Distribution of Replay Occurrences - Weekly ......................................... 37
Figure 4-15 Time Distribution of Replay Occurrences - Daily ........................................... 37
Figure 4-16 Simultaneous video downloads within time margin ......................................... 38
Figure 4-17 Overall network gains in regard to categories ................................................ 40
Figure 4-18 Weekly network gains in regard to categories ................................................ 40
Figure 4-19 Network Gain Trend ........................................................................................ 41
Figure 4-20 Overall terminal gains in regard to categories ................................................ 42
Figure 4-21 Weekly terminal gains in categories ................................................................. 43
Figure 4-22 Category contributions to the overall network gain ......................................... 44
Figure 4-23 Network gain and Net gain in regard to categories .......................................... 45
Figure 4-24 Network Gain & Net Gain weekday distribution ...........................................46
Figure 4-25 Network gain Hour Distribution of all Music traffic ........................................47
Figure 4-26 Network gain of video traffic with different durations ........................................48
Figure 4-26 Network gain of video traffic of categories with different durations .................49
List of Tables

Table 3-1 An extract from the main table in data warehouse ................................................................. 24
Table 3-2 an extract of index and hashed MAC ...................................................................................... 25
Table 4-1 Weight-added network gain ................................................................................................. 44
List of Acronyms and Abbreviations

API Application Programming Interface
BRAS Broadband Remote Access Server
CAPEX Capital Expenditure
CDF Cumulative Distribution Function
CDN Content Delivery Network
CTTL Cache Time-to-Live
DoS Denial-of-service
DPI Deep Packet Inspection
DSL Digital Subscriber Line
DB Database
DW Data Warehouse
FTP File Transfer Protocol
FTTH Fiber-To-The-Home
HTTP Hyper Text Transfer Protocol
IoT Internet of Thing
ISP Internet Service Provider
MAC Media Access Control
OPEX Operational expenditure
P2P Peer-to-peer
QoE Quality of Experience
QoS Quality of Service
SQL Structured Query Language
VoD Video on Demand
Chapter 1 Introduction

1.1 Overview

1.1.1 Video Streaming Traffic Explosion

User Generated Content (UGC) has become popular since the birth of different web services allowing the distribution of such user produced media content, which represents one of the greatest changes of web services starting from early 1990s [1]. Today UGC on the Internet varies from textual information in Blogs, photo galleries like Flickr and Facebook to videos such as YouTube and Vimeo. Cisco Visual Network Index (VNI) forecasts that by the end of 2012 Internet video streaming traffic will consist of over half of all Internet traffic and by 2015 Video on demand (VoD) traffic is going to triple, equal to 3 billion DVDs per month [2]. For only YouTube, currently the most popular video-sharing website, it accounts for about 60% of all videos watched on the Internet with 65,000 video uploads per day according to estimates [3].

Apart from this, the successive popularity of Web 2.0 and Internet of Thing (IoT) also push the growth of media streaming traffic [1]. Especially Web 2.0 changes how users participate in the Web unconsciously. Since it is designed in such an easy way for users to sign in and post contents of any format other than viewing contents [4], a tremendous amount of UGC traffic has been generated passing through the Internet. Besides, HTML 5 today has come upon the stage of the IT world [5] with quite many new features related to a diversity of web applications that will generate considerable volume of video/audio streaming traffic. For example, WebSocket protocol, which enables a bidirectional ongoing conversation between a browser and a server while keeping the TCP socket open [6], facilitates the live content delivery and provides a highly flexible interaction between the end user and the Internet.

A more concrete view of how large the video streaming traffic has exploded is illustrated by Cisco VNI in Figure 1-1 [7]. The outburst can be clearly observed that by the year of 2016 the total Internet traffic is going to be 4 times larger than the traffic amount with an incredible global video traffic leap up to about 30 times. Given mobile traffic, the total volume of mobile video communications in 2014 tends to
increase over 5 times than that in 2011. YouTube, accounting for 52% of global video streaming traffic [8] predicted by Allot Communications [9] and Netflix, making up nearly 30% of all downstream traffic [10], are two main drives for this video explosion.

![Figure 1-1 Video traffic explosion of global traffic and mobile traffic](image)

However this global video traffic booming is a double-edged sword. On one hand, it shows how user behaviors have changes and what user interest trends to be, which promotes the development of new applications as well as accelerates the evolution of the Internet. On the other hand, the continuously growing traffic traversing through the backbone networks is going to severely strain the network resources on both centralized servers and branching network links and meanwhile incurs high cost for Internet Service Providers (ISPs).

Because the increasing operation and maintenance cost gives rise to a large figure of operating expenses (OPEX), ISPs start to look for the ways of either increasing capital expenses (CAPEX) or lower the OPEX on condition that the same quality of experience (QoE) of end users is guaranteed. Figure 1-2 collects the statistics from Yankee group, revealing approximately 1.5 times growth in ISPs’ revenue from 2010 to 2013 [11]. Since the video streaming traffic is eating up a massive proportion of bandwidth but bringing in a low increase in revenue, the issue of efficient content distributions and video streaming traffic optimization has become a hot research topic in the past few years. Subscribers seek for quick online video delivery and smooth video content continuation while network operators focus on reducing the network load and increase earnings per bit of videos.
Currently there are several methods to address this problem for content providers. The most widely applied solution is to implement content delivery networks (CDNs) to intelligently distribute contents with distributed caches set on CDN servers. Figure 1-3 reveals the overall network architecture of a typical CDN, where the published contents are tactfully spread out into the meshed networks that are close enough to end users with fewest hops. In this manner, it technically solves the problem of long response time when users try to retrieve large sizes of contents caused by limited bandwidth, high user hit rates, or misdistributions of data center nodes. Even the largest video sharing site YouTube uses CDNs to distribute its most popular contents since creating its own network would spend a long time and a high cost [12]. However, this technique has several drawbacks. First of all, the content provider has to pay the CDNs for their services, which could turn out a waste of investment if the contents are not as popular as expected. Secondly, which kinds of principals are applied to decide whether this content should be distributed in advance of retrievals to the CDN is another key issue. Last but not the least, a high scalability of the network topology is hard to achieve and the cost for upgrade is larger compared to other techniques.
1.1.3 P2P

An alternative to CDN is to exploit Peer-to-peer (P2P) techniques which give a simple content distribution solution with a low implementation cost. Today P2P is widely used by users for file sharing with its abundant resources. From Figure 1-4 [13], P2P traffic accounts for more than a quarter of all bandwidth in Europe, and 40% of all packets sent. Among all P2P traffic, BitTorrent takes up an outstanding portion. Spotify, one of the most popular and successful Swedish music streaming services, uses both a BitTorrent-like tracker to scale up demands from millions of users and a Gnutella-style network [14] but is specifically tailored towards relatively small files. Figure 1-5 shows that only 8.8% of all music playbacks originate from Spotify servers. The rest come from the peer-to-peer network (35.8%) or the local cache (55.4%). PPLive is one of the most popular Chinese applications based on P2P technology, providing both live video streaming and VoD services [15]. The drawback of P2P technique, however, is that P2P protocols are unaware of the physical network topology, resulting that end nodes generate a great deal of inter-ISP traffic, among which much traffic is unnecessary due to the blind routings back and forth between ISP networks. Besides, no central node exists in P2P network and each node behaviors depending on itself. Thus the effective user management could be hard to achieve and Quality of Service is not guaranteed.
As has been seen from Figure 1-5, streaming traffic fetched from local cache can take up over half of the total amount of Spotify traffic, indicating that various forms of local cache are feasible solutions to reduce the transit traffic as well as enhance the performance of services. Caching got a high attention and was widely deployed when the world-wide web emerged. These days it becomes one of the top hottest issues again and the phenomenon of cacheability has been discussed in the P2P network community [16]. YouTube and BitTorrent both suggest that it is time now to further look into caching [17]. The caching technology has proved to be a vital way in future to cope with the bandwidth constraints in ISP networks for both PC and mobile users [18].

1.2 Goals

Summarizing from all issues listed above, analyzing traffic traces in live IP-access networks in order to gain knowledge of content demand patterns and
cacheability is put forward of a high importance from perspectives of both ISPs and subscribers. Parameters of interest are e.g. user-preferred applications, traffic load, usage patterns, hit ratio, network gain etc.

1.2.1 Project Motivation

This thesis project is one part of IP Network Monitoring for Quality of Service Intelligent Support (IPNQSIS) Project [19], which focuses the customer perception and network performance as the main drivers for building a complete Customer Experience Management System (CEMS).

Due to the previous papers reflecting the explosion of video streaming traffic, my thesis project lays the concentration of monitoring YouTube traffic and investigating YouTube content demand patterns and cacheability. All the data analyzed in the thesis project is collected from a Swedish commercial municipal network and is analyzed in a test-bed monitoring network in Acreo.

1.2.2 Thesis Project Goals

This master thesis project aims at finding potential interesting YouTube content patterns and the possible high cacheability given video clip metadata under a Swedish municipal network. It also aims at confirming some universal knowledge for future researches. It is achieved by first seeking to develop a measurement framework to associate cacheability with the metadata knowledge of video category and video duration. Then based on the framework, experiments on parameters of cacheability are carried out to investigate potential interesting laws of content patterns and user behaviors.

Specific tasks are concreted out, involving the setup of this measurement framework as well as analysis on contend demand patterns of YouTube and parameters of cacheability. Detailed YouTube traffic traces and packet monitoring are investigated. Massive data management and statistics analysis are carried out as a systematic procedure on data manipulation with the help of MySQL database and Python/PHP API scripts.

1.3 Project Scope

The scope is to construct a measurement framework under the Acreo network environment. The amount of YouTube traffic flows that can be freed up by introducing a smart cache strategy who knows video metadata information is measured. YouTube streaming traffic patterns are traced and potential cacheability laws are investigated.
A commercial municipal network, as an important composition of the Internet has a mature subscriber base and the services they choose. Subscribers tend to be independently distributed. Each of them has his own targets and behaviors. Besides, no general regulations concerning video downloading or playback are applicable as compared to those of campus network or enterprise network. With totally different access-speed subscriptions provided by competing ISPs, users vary their network services subject to the bandwidth.

A specified measurement framework is setup to find potential cache gains in regard to video clip information, based on which content patterns of YouTube video traffic are studied and compared with user behaviors. The cacheability is mainly analyzed with certain parameters such as network gain, hit ratio, terminal gain, etc based on the monitoring data filtered from the entire YouTube traffic passing through the network in order to gain knowledge of the potential lifetime and size of various types of caches. The metadata of each video is studied as well, based on which each end user is categorized with multiple labels. Video duration is also associated with network gain to find more interesting results.

Therefore, the target audiences of this report are mainly small or medium operators of IP-access networks who wish to reduce resource consumption on backhaul links. Researchers working in the domain of network cacheability of online video streaming are included as well. People who have fundamental knowledge of access network cache strategies are also welcome to read it as a reference.

### 1.4 Expected Outputs

This thesis project intentionally brings network cacheability together with the video clip information by constructing a generic measurement framework based on the network environment. This new measurement framework should be adaptable for multiple media content provider sites and the parameters of interests can be configured accordingly.

Then based on it experiments are performed with a focus of various parameters. Traffic traces as well as network cacheability with different video categories and video duration are supposed to disclose interesting laws, which will be further analyzed. The overall network gain is considered high if video clips can be cached based on their categories or durations. Moreover, another expected output is to show the necessity of an intelligent cache engine that can make a decision on each video category or duration information based on my conclusions.
1.5 Report Composition

The thesis report consists of six chapters: Introduction, Backgrounds, Methodology, Results & Analysis and Conclusion & Future work.

Chapter 2 discusses the background of the research and some necessary concepts that are applied in the project. The previous work from both the global range and within Acreo is also listed along with the hypothesis.

Chapter 3 reveals the approaching procedures of data parsing and the overview of network architecture. Methodologies are described in details to make readers well understand the network environment and how the measurements are carried out. The measurement framework is illustrated with the data warehouse structures.

Chapter 4 describes the experiments and analyzes the results achieved. Some cacheability phenomenon discussions and peculiar user behaviors are taken into consideration.

Finally conclusions are drawn in chapter 5. Summaries are made on what has been done and what is considered left to be done. The thesis ends up with a short discussion of any possible future improvements along the track of this work.
Chapter 2 Backgrounds

This chapter describes the essential background knowledge of this thesis work. Section 2.1 briefs some research results of the traffic usage and user behaviors with a Quality of Experience (QoE) discussion. With the summary suggesting a high potential benefits that could achieve by enabling an optimal caching mechanism in the network, section 2.2 focus on the introduction of locality-aware networks and the cacheability and section 2.3 describes the aggregated network patterns. Next, we move on to present YouTube development and its strategies in section 2.4. In the end all relevant previous researches in the global and within Acreo are listed in section 2.5.

2.1 User Behaviors and QoE

In order to optimize the network design and improve user service delivery, the traffic patterns and service usage should be comprehensively examined and fully understood. The investigations of user behaviors reflected by traffic models analysis and Quality of Service (QoS) have been performed in a wide range by academia. The Traffic Measurements and Models in Multiservice Network (TRAMMS) project did the researches of the traffic characteristics at aggregation levels in multi-service networks in Spain and Sweden from 2007 to 2009 [20], in which a traffic monitoring framework is utilized to gain a deep insight into IP traffic patterns, especially for that of video streaming applications. The IP Network Monitoring for Quality of Service Intelligent Support (IPNQSIS) project is a successor to the TRAMMS project [21] which focuses on investigating network performances regarding user behaviors and QoE to build a Customer experience Management System (CEMS).

As is known to all, user experience is a vital criterion for service providers. They deploy various measurements of QoS for their users and intelligently design their business strategies according to those results. A great deal of researches have been done using parameters to characterize the traffic patterns, for example the overall traffic volumes daily, weekly and monthly in [22], the composition of applications or protocols in [23], and traffic monitoring over time scales for specific applications such as Spotify and Voddler [24].
Quality of Experience (QoE) has been referred to as a metric for evaluating the Internet services as a whole [25]. It is defined much dependent on users’ comprehensive subjective feelings towards network performance, system equipments, end-to-end QoS, etc [26]. As the users’ perceived experiences towards application layer presenting overall results of the individual QoS, the QoE indicator has raised high attention among service providers in recent years. The Telecommunications Management (TM) Forum [27] is therefore founded comprising over 220 world’s leading service providers, aiming at promoting the service providers’ business with a low complexity. Their technical report: Managing the Quality of Customer Experience [28] examines how to measure user experience by studying a wide range of delivery mechanisms in real cases. In order to meet all requirements listed in the technical report, a holistic end-to-end user experience framework consisting of six application programming interfaces (APIs) are proposed [29]. They are used to keep track of QoE metrics and also predict the trends of user behaviors. The importance of knowing customers’ relationships is also emphasized to further enhance customers’ satisfaction in the report.

2.2 Locality-Aware Networks and Cacheability

Since VoD and P2P traffic keeps increasing the consumption of bandwidth in the Internet, lots of researches have been performed and various methods to reduce traffic between ISPs have been proposed. Caching the contents with a high hit ratio is considered of high efficiency as well as low costs to reduce the inter-ISP traffic. Much of work about the impact of caching and the evaluation methods in different systems have been carried out [30-34].

A normal web service caching could be put in either the user terminal or in a proxy server. A terminal cache is usually stored in the web browser temporarily for the potential future use and is released if the memory is full or is manually removed by users. A proxy cache is located between a client and a web server where a local copy of the contents that is likely to be frequently requested in the network is stored [35]. It offloads the burden of the actual web server.

Other than using a local cache or a proxy cache, users in the same network might take advantage of each other’s local cache as a resource to search for his interesting contents. This has already been implemented in P2P networks. Despite of the drawbacks of P2P mentioned in Chapter 1, a locality-awareness P2P network can improve network efficiency by clustering users geographically. However, in many local networks users tend to be independent on the interests, such clusters do not seem to achieve a high gain given the deployment costs. Thus then other kinds of user clustering are proposed based on the video metadata such as video category and video duration. Given such knowledge aware by the cache [31], a more flexible and efficient
caching mechanism can be achieved with a high cacheability to dramatically reduce the streaming flows on backhaul links.

Theoretically a piece of content is regarded as cacheable if it has been retrieved more than once. According to Ager, et al., the cacheability of n items can be calculated by the following formula [36]:

\[
\text{Cacheability} = \frac{\sum_{i=1,n}(t_i - 1) \cdot s_i}{\sum_{i=1,n} t_i \cdot s_i}
\]

\(t_i\) indicates the total downloading times for item \(i\) of which the size equals \(s_i\).

Ager, et al. found that the cacheability for P2P applications is 27% considering only local hosts but only less than 10% when it comes to UGC sites [36]. One explanation for the low figure of UGC sites would be that the content is only allowed to be cached if the proxy obeys a correct cache-control header, which P2P application does not need to follow. Leibowitz et al. [37] took a deep look into P2P cacheability by measuring the caching gain in terms of byte-hit-rate. Their results indicates that higher traffic volumes yield higher caching gain and 67% of the bytes can be served from the cache – even use a cache of only 300 GB. Zink et al. [38] analyzed YouTube traffic and proposed three kinds of caching structures which compares the caching performance on a simulation of a large university campus network.

2.3 YouTube

With 4 billion video views per day and one hour of video uploads per second [40], YouTube has undoubtedly become the key platform for video streaming sharing worldwide. New viewing patterns and social network interactions are created, which draws lots of interests for researches. Besides, YouTube also acts as a fresh way to boost personal visibility or broadcast products. [41] A great deal of market interest inside YouTube has been added.

2.3.1 YouTube History

YouTube was founded by 3 former PayPal employees on Valentine’s Day in 2005, whose initial goal was to upload and share videos among friends. The name and logo were both inspired by cathode-ray tubes in televisions. In the following year, YouTube welcomed a hugely popular growth and therefore purchased by Google in November 2006, with 1.65 billion US dollars in stock [42]. Since 2010 YouTube introduced the online-film rental service, offering over 6000 films. Meanwhile, the website is on its way of improving the user interface design.
2.3.2 YouTube Today

Today YouTube website is available with localized versions in 42 countries, supporting 54 different languages [42]. From Figure 2-1 we can find that nearly half of the video streaming traffic passes through Google sites.

In addition, the user interface has been designed more simplified for subscribers to easily find out their interested channels. Recommended videos are thoughtfully sorted by YouTube based on the subscribers’ history list. More statistics of videos can be accessed through the page when subscribers watch them. Furthermore, YouTube has removed the limits of the maximum length of uploaded videos. This leads to a burst of movies whose lengths reach over 30 minutes during the past several months.

Since any registered user can upload unlimited number of videos, subscribers can set up their private cinemas, news release channels or amateur video albums, which gradually replace the traditional media. From YouTube, they need to consider how to successfully handle millions of uploaded video clips everyday and provide better user experiences.

<table>
<thead>
<tr>
<th>Videos (000)</th>
<th>Share of Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Internet : Total Audience</td>
<td>201,420,689</td>
</tr>
<tr>
<td>Google Sites</td>
<td>88,278,970</td>
</tr>
<tr>
<td>Youku Inc.</td>
<td>4,644,727</td>
</tr>
<tr>
<td>VEVO</td>
<td>3,697,229</td>
</tr>
<tr>
<td>Facebook.com</td>
<td>2,590,612</td>
</tr>
<tr>
<td>Dvango.co, Ltd.</td>
<td>2,488,180</td>
</tr>
<tr>
<td>Tencent Inc.</td>
<td>2,343,437</td>
</tr>
<tr>
<td>Tudou Sites</td>
<td>2,321,239</td>
</tr>
<tr>
<td>Sohu.com Inc.</td>
<td>2,290,461</td>
</tr>
<tr>
<td>Hulu</td>
<td>2,104,724</td>
</tr>
<tr>
<td>Microsoft Sites</td>
<td>1,830,052</td>
</tr>
</tbody>
</table>

Figure 2-1 Share of videos in popular sites

2.3.3 YouTube Strategies

In YouTube, Sorenson Spark and Adobe Flash 9 are used as the video coding technology. The video playback is achieved through Flash, which supports over 75% online video formats and is a quite common plug-in for today’s computers. Because of the low bandwidth requirements and the simplicity of being embedded in other websites, YouTube wins over lots of applications that use online streaming
techniques.

Since Jan 2011, YouTube canceled the maximum video length limit. Movies with long durations crowded into all channels with a higher resolution. Meanwhile the upload interface was upgraded and it supports multiple uploads at the same time.

From the most original H.263 video codec and to current H.264/MPEG-4 AVC and stereo AAC audio supporting standard quality (SQ), high quality (HQ) and high definition (HD) to near future’s 3D/4D stereoscopic contents, YouTube never fails to satisfy subscribers by continuously enhancing video quality and user viewing experiences. In the VidCon 2010 meeting, YouTube claimed that it started supporting 4K videos (videos with a resolution of 4096*3072) [43].

The basic YouTube video retrieval process is depicted in Figure 2-2, which typically has two main phases: content look-up and content download/playback. First of all, the user sends out a GET request to retrieve the video content. YouTube web server redirects the request to the nearest content delivery network (CDN) server based on the video ID and YouTube mechanisms if no content is available on the web server. Then the user resolves the server name and sends out a new request with the string ‘videoplayback’. The CDN server answers it with a usual 200 OK response, starting the content downloads for each video clip before the user requests for the next part.

![Figure 2-2 Basic YouTube video retrieval](image-url)
For PC users, an optimized mechanism is applied under the condition that the web browser has enough cache during the second phase. For example, the aggressive buffer policy initiates a fast delivery start regardless of the actual watched length of the video. However, the mobile users, which are considered as a client with a smartphone, a tablet or a set-top-box with some applications, have limited terminal cacheability and lots of unoptimized video streaming processes increase the delay of contents delivery during the second phase.

2.3.4 YouTube Data API

YouTube has a data API that integrates YouTube functions into developers’ own apps or websites [44]. It supports Java, JavaScript, PHP, Python and .NET for the client library, which facilitates the data query and retrieval without manually generating HTTP requests or dealing with HTTP responses.

The below figure shows the main structure of how the workstation is connected to YouTube API and fetch data to store in the local data warehouse.

![Figure 2-3 Data Parsing from YouTube data API](image)

2.4 Previous Analysis on YouTube Traffic

2.4.1 Prior Researches

Several researches and papers concerning various content and behavioral properties of VoD traffic are already available, but none of them has analyzed the traffic traces and content demand patterns based on the video category and video duration as in this project, which concentrates much on the user behaviors and corresponding necessity of locality and caching strategies in the set of video streaming traffic with the same category and similar durations.

[61] offers a case study of YouTube short video sharing characteristics
comparing statistics ranging from access pattern to active life span. The social networking aspect of YouTube is also examined. In [62] the impact of YouTube traffic on an ADSL platform of a major ISP in France is characterized. The conclusion is drawn based on the YouTube server volumes and throughputs. As an extension of this work, a long term analysis of YouTube characters is figured out as well. [38] and [45] studied YouTube strategies and investigated user generated contents to draw their conclusions, that is for example, over 18% of the time a video requests redirects. From [38], no strong correlation between local and global popularity of YouTube videos enables us to independently analyze YouTube traffic patterns in each certain area. Neither time scale nor user population is shown to have a significant impact on the local popularity distribution of video clips served by YouTube.

User behavior and implication investigations in [46] reveal that 50% of resolution switch from low to high happens in the first 10 seconds and over 80% of the switches happen in the first 20% of the video length. Moreover, 60% of videos are watched for less than 20% of their duration on both PC and mobile users, which hints us that caching only some portion of videos would be a good solution. Considering user early abortion, for PC 40% of sessions downloaded two times more than the amount of the data that was actually watched, while for mobile users 20% of the sessions downloaded 5 times more than the amount of watched data.

2.4.2 Previous work in Acreo

From [47] issued by Acreo, the streaming-media services like YouTube, online TV etc. are the driving force for the increase of total network traffic while the dominant traffic volume of P2P file sharing remains the same. According to this, the focus of my thesis work lies on the user generated traffic of streaming-based video/audio services and the potential cacheability that could be achieved.

Yichi Zhang looked into aggregated traffic patterns as well as two household traffic models [48], which provide some practical analytical methods for our experiments to refer to. But the user identifier is based on IP address, which varies every certain amount of time, introducing inevitable errors when the same IP address is leased to another end user. In this project the MAC address with a specific user-agent string is utilized as a unique end user identifier.

Manxing Du did analysis on the network hit ratios and potential cacheability of different geographical regions on the YouTube traffic. However, due to the lack of metadata, her results are restricted in the basic video retrieval investigations instead of combining user behaviors with the video content patterns.

This thesis project is greatly motivated by the earlier researches and turns out a continuation on YouTube traffic patterns and user behaviors in Acreo as one part of IPNQSiS project, which was launched to develop the mechanisms that would allow
monitoring and managing services offered on the next generation IP networks, such as IPTV, Mobile TV and VoIP from the customer perspective [49]. Permission to investigate subscribers’ data was obtained. The physical resource for carrying out investigations is provided uniquely by Acreo and confidentiality agreement has been signed.
Chapter 3 Platform and Methodology

This chapter illustrates the overall network platform on which my project measurement framework is based and the whole approach of the thesis project. The utilized methods are described in detail.

Section 3.1 reveals the main approach of the thesis project. The network architecture can be found in section 3.2 as well as the network patterns in section 3.3. Methodology of data collection and data analysis is depicted in section 3.4 and section 3.6 respectively. Section 3.5 represents the data warehouse constructs.

3.1 Main Approach

As Figure 3-1 depicts, the general approach of the project consists of five parts: preliminary work, implementation, results analysis, verification and conclusions. The preliminary work includes literature study, theory researches and prior work. Based on the estimated outputs some further researches are carried out to obtain a feasible method for real implementation. After the results are gathered, verification methods are deployed to see if they make sense. If not, certain part needs to be modified or even the whole method is supposed to be reconsidered. Finally, conclusions are drawn from all these testified results and related future work is proposed.
If the validity and veracity of the results fail to meet the criteria, certain improvements are required and implementations with the new improvements are carried out again. The validity checking could be based on empiricism, prior results, reference materials, etc. Improvements that have been made are parameters-oriented.

### 3.2 Network Architecture

Basically the whole project is carried out on two networks as seen in Figure 3-2: the Municipal network where all subscribers’ raw data is captured and the test-bed network, to which the data warehouse is connected to.

The PacketLogic server is connected to gather raw data from the municipal network and store it in the data dump database server. The data dump database server has a 2-TB disk memory which is enough for store at least one-month raw data in our experiments. To protect the user privacy, no raw data is allowed to leave the municipal network. All data on the data dump database server must be parsed and extracted to get rid of all sensitive subscribers’ information before stored in the data warehouse in the test-bed network. The parsing process is carried out with python scripts.
In the test-bed network, the data warehouse server is the place where all parsed data is stored to be further analyzed. The data is post-processed here before put into tables in the MySQL database by various python scripts based on our interested parameters. The MySQL database is set on a Ubuntu Linux distribution server, which can handle all kinds of data manipulation in this project.

3.3 Aggregation Network Patterns

The emergence of triple play services has made the need for deployment of Ethernet aggregation networks prevalent. These networks must be able to deliver emerging multimedia applications. Deployment challenges of Ethernet as an aggregation technology are both technical and economical. Studies have shown that a single platform capable of operating as an Ethernet aggregator, Broadband Remote Access Server (BRAS), and a Provider’s Edge (PE) router can provide the optimal solution technically and economically [39]. Technical factors to consider in construction of such network are traffic management, high availability, network topology, scalability and security.

The municipal network in this project has around 2000 households. Each household takes the 1Gbps link to its access network, from where the traffic is aggregated to 10Gbps Internet gateway through aggregation network. As Figure 3-3 reveals, the fine line denotes 1Gbps links while the bold lines represents 10Gbps links. The measurement probe is connected to the Internet edge via optical 50/50 splitters which takes the optical signal into 2 exact copies, one for measurement and the other for transmitting to and from network devices. As the measurement probe is configured passive it works independently and does not affect the network traffic.

![Figure 3-3 Municipal network architecture](image)
3.4 Data Collection Framework

On the municipal network over 60% of the video streaming traffic are generated from YouTube site according to the prior work in Acreo. Since our network consists of approximately 2200 households and 6800 end devices, the chance is high that the same video content is streamed into the municipal network several times, wasting a great amount of network resources. Thus, we track the YouTube ‘GET’ request signaling sent from the network to analyze the user behaviors.

In the ‘GET’ request signaling the YouTube video ID can be retrieved. It is an identifier of each video given by YouTube. We use video ID to collect video information from YouTube API and store it as the video metadata.

From Figure 3-4 it is clear to see how the whole data structure is organized before sent to the data analysis engine. In the following section 3.3.1 we focus on the ‘GET’ request signaling collection methodology while section 3.3.2 describes the video metadata collection.

3.4.1 HTTP Request Data Collection

Since we are ‘sniffing’ the municipal network at the network border and dump raw data generated inside the network, the goals in the signaling collections are set first to make it secure and efficient:

- Collect only the valid request signaling sent out towards YouTube sites for starting a video streaming download.
- Gather such data for an extended period of time. In this project, the dataset contains at least three-week data.
- Protect user privacy.

In the data dump, we only consider the outbound requests which aim at starting a video streaming download. On the PacketLogic client the filtering rules are defined as we only collect HTTP GET requests with a standard YouTube-identified string. The
challenge is that some YouTube GET requests might be missing due to the encryption of the whole URL by enabling tunneling or HTTP Secure mechanisms while a few requests could be faked, resulting in no actual video downloading because of, for example the denial by the YouTube server or no content available for the link. The challenge will affect the final result within an error deviation controlled lower than 5% according to the proportion of the requests containing missing data in total requests.

We keep data dump for at least a month and the data is checked everyday to guarantee no technical problems pop up as well as to verify the data reliability, including the timestamp when each request starts, the date/time when one pcap file is full, the time consecutiveness of the requests, etc.

To protect the user privacy, all the data collected is forbidden to leave the municipal network as Figure 3-2 reveals. The YouTube visitors’ identifiers that are fetched from the HTTP header lists get hashed before they are transferred into test-bed network. They are afterwards stored in a MySQL table identified by their indexes. Thus in future analyses each end user is presented by a unique integer. The hashing function is a standard python MD5 digest algorithm combined with an arbitrary update string. The update string is settled once the first data parsing is executed and remains fixed for the same collection of data to be parsed.

The user privacy is also assured by putting the MAC address of the family and the user agent string together immediately when the data is parsed before calling the hash function. In this way, even if someone sniffs into the data during the parsing process, he could only get the messed string, which has no way tracking back to the user information.

3.4.2 Metadata Collection

As Figure 3-3 displays, the YouTube video ID data, which is collected as section 3.3.1 describes, is sent to YouTube API to retrieve video metadata, that is, the video information such as video duration, category, view counts, favorite counts, ratings, etc. For each HTTP GET request, the video metadata is collected via YouTube API based on the video ID and put into the same row in the MySQL database table.

The metadata collection is done with PHP scripts since the XML file that YouTube API returns get quickly parsed through PHP scripts and the child objects are easily changed if the interested parameter alters.

Challenges occur during the implementations. When we keep the inquiry towards YouTube API for a long period of time, it detects that massive requests are received from the same node and blocks the socket based on its defensive mechanism to avoid the potential denial-of-service (DoS) attack. Therefore, in the implementation process we have to manually restart the PHP script on a new connection every certain amount of time in order to deceive the YouTube API not to start its anti-DoS attack
mechanism.

Some video IDs return empty metadata in all fields due to the unavailability of the content on the website. The reasons for this are mainly the following:

1. It is a private video. Access is denied.

2. The video content no longer exists. It could be removed by the uploader or the account associated with this video has been terminated.

3. The video ID is wrong. Human typing errors often happen when users manually type in the URL.

The number of requests with empty metadata accounts for less than 5% of all requests in the dataset. Since these requests are generated by users, they are valid when we consider the total data amount. But they are not taken into considerations when it comes to the analysis of video metadata parameters. After evaluating the side effects that the empty metadata could bring to the final results, we decrease the error as much as possible during experiments.

3.4.3 PacketLogic – Data Collection Tool

PacketLogic is a commercial network packet inspection and capture tool, aiming at providing deep packet inspection (DPI) and optimizing bandwidth for telecom operators, post, telephone and telegraph administration (PTT) and research institute [52]. This tool has an outstanding user interface visibility and convenient configuration, which facilitates management of the whole system. Furthermore, it is powerful on data collection and personal rule settings. It is a rack-mounted system which can be equipped with many kinds of interfaces [53].

In our network, PacketLogic server is put on the measurement node between the aggregation router and the Internet edge. Since two redundant links are added to the node, two physical GB Ethernet channels are deployed.

The workstation installs the newest version of PacketLogic client application to set filtering rules and data dump regulations. It also has a Python API that can enable user-defined programs for system configurations. All frequently used features can be accessed. For example, configuring objects and rules sets to adjust the monitoring criteria to meet the operator’s needs, browsing real-time traffic data through Live View module, displaying statistical charts in the Statistics module, etc.

3.4.4 Data Filtering and Parsing

From the above the data collection methods are listed. But how to efficiently extract time-stamped data from the database and put it into the warehouse without introducing dirty data or unnecessary data is a tricky procedure. In this project, it is done by parameter-tailored python scripts as Figure 3-5 shows.
In the scripts, a basic format of the output data structure is initialized. Each data packet is parsed then and useful fields are stored in the corresponding output fields of the data structure saved as a txt file in the data warehouse. Hash function is executed to encrypt sensitive user data like MAC addresses. User agent string mapping function is carried out in the scripts to obtain the end device type as well as the browser information.

### 3.4.5 IP-MAC Mapping Scheme

The municipal network is a layer three network. Therefore the MAC address in the packet changes hop by hop. The source MAC address in our data dump is always the last hop router’s MAC address, which reveals no information of the end household/device. Since the PacketLogic server only dumps the packets which contain the timestamp and the source IP address which is temporarily leased to the subscriber, the researches need the unique end user identifier. That is to say, we need to fetch the constant MAC address of each subscriber. Thus, another database table which stores the corresponding IP-MAC addresses matching information every five minutes is applied. This table extracts data from DHCP logs and converts it into new entries for all in-used IP addresses every 5 minutes. Python scripts are used in this scheme with a library that has been made ready for mapping the MAC address with the already known timestamp and source IP address. The data dump server has 4 CPUs to support running all the mappings.

### 3.4.6 User Identifier

In the experiments the end user is identified by his encrypted household MAC address string plus his hashed user agent string. This user identifier string is unique for each end device and the chance that two end devices in the same family have the exact same user agent string is low. In this way, the end user sub-table is built and the requests that are sent from each end user are linked.

### 3.5 Data Warehouse

Data warehouse is precisely designed for massive data analysis. Unlike database, which is transaction-based and focus much on capturing real-time data, data
warehouse enables further data mining with strict date/time attributes and multiple data sources. Once it is setup, the data is rarely modified except for adding new data.

To make it better serve this project as well as my report, a high capacity of redundancy is introduced in the data warehouse. Thus, a large storage memory is required for each subject in the warehouse.

A good warehouse should be designed to work efficiently. Besides, the data filtering and collection should be accurate. The time scalability is also an important factor to make the warehouse work stably [54].

The following sections introduce the hierarchical layers of tables in the data warehouse server as well as their content. Section 3.4.1 describes the main table and 3.4.2 shows other sub tables. Section 3.4.3 provides a whole view of how these tables are joined and how the shared fields are mapped.

### 3.5.1 Main table structure

After data collection is finished, the tables in the data warehouse are created and all original statistics is stored. A main table stores all the data with the index of each field in sub tables, which is covered in section 3.4.2.

<table>
<thead>
<tr>
<th>id</th>
<th>Datetime</th>
<th>Mac_hash</th>
<th>User_age</th>
<th>User_t</th>
<th>Video_</th>
<th>Categ</th>
<th>Durati</th>
<th>View_C</th>
<th>Favorite_</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2012-06-08 16:32:30</td>
<td>345</td>
<td>127</td>
<td>PC</td>
<td>31</td>
<td>2</td>
<td>34</td>
<td>176654</td>
<td>5433</td>
<td>4.94</td>
</tr>
<tr>
<td>2</td>
<td>2012-06-08 16:32:32</td>
<td>35</td>
<td>326</td>
<td>Mobile</td>
<td>4532</td>
<td>4</td>
<td>134</td>
<td>22543</td>
<td>23</td>
<td>4.21</td>
</tr>
<tr>
<td>3</td>
<td>2012-06-08 16:32:38</td>
<td>67</td>
<td>1327</td>
<td>TV/Pla</td>
<td>23112</td>
<td>1</td>
<td>12</td>
<td>72983</td>
<td>12</td>
<td>4.30</td>
</tr>
<tr>
<td>4</td>
<td>2012-06-08 16:32:43</td>
<td>1432</td>
<td>897</td>
<td>unknow</td>
<td>438</td>
<td>8</td>
<td>8210</td>
<td>129807</td>
<td>1536</td>
<td>5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

*Table 3-1 An extract from the main table in data warehouse*

The fields of main table contain subscriber information about index, date and time, hashed MAC address, hashed user agent string, user type, video ID dumped from the municipal network and video information about video duration, category,
view counts, favorite counts and rating obtained from YouTube API. Each row of the table represents a HTTP GET request to start a YouTube video content retrieval.

Some rows have empty field of the video metadata but we still regard them as valid request from the network. It also contains unknown user type which we can get no such information from the user agent string that matches our user type table. Some family might deploy some kinds of TV systems which cannot be distinguished, for example, some new playstation 3 systems.

### 3.5.2 Sub tables structure

From Table 3-1 some fields only include integers which are referred to another sub table. The sub tables facilitate the search functions and take great advantages of the joint function. Besides they make the data warehouse structure more concrete and readable. In regard with the main table, sub tables are included such as hashed MAC table, hashed user agent table, video ID table, Category table, etc. Besides, the IP/MAC mapping sub table as described in section 3.3.2 is covered.

Table 3-2 gives an example of the sub table about index and hashed MAC. The id of each corresponding hashed MAC is referred in the main table. Only the id in each sub table is used in future investigations.

<table>
<thead>
<tr>
<th>id</th>
<th>MAC_hashed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>f4b2f476310d96953393ada0bf72142c</td>
</tr>
<tr>
<td>2</td>
<td>83586badd33a21a255ff567c2602e071</td>
</tr>
<tr>
<td>3</td>
<td>af0b4c7eab89a74065874152944bebea</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 3-2 an extract of index and hashed MAC

### 3.5.3 Join structure among tables

The join function is applied to connect the main table with the sub tables by referring the index.
From Figure 3-6, a column named id in the sub tables gives each row an identifier, which is used in the main table. The join function of MySQL is applied to map them with a simple query ‘Main.mac_hashed=mac.mac_hashed’.

### 3.6 Data Analysis Method

#### 3.6.1 Analysis Tool

According to the dataset stored in data warehouse, we try to extract certain video metadata patterns and find possible interesting features from the huge amount of traffic with data mining methods. Several analysis tools are adopted in the project.

Python, as one of the most popular scripting language, has a large standard library with a high code-readability. Therefore it is majorly utilized during the whole project. Python also embraces a large variety of powerful APIs, which are usually known as modules in Python. In our analysis processes, the python MySQL API is used to communicate to the data warehouse [55]. In the python scripts, the following statements can be used to create such connection to a MySQL database server on our workstation provided the use credentials and database name.

```python
import MySQLdb
MY_HOST = '192.168.x.x'
MY_USER = 'xxx'
MY_PASS = 'xxx'
MY_DB = 'xxx'
con = mdb.connect (host = MY_HOST,
                 user = MY_USER,
                 # other arguments as necessary
                 )
```
passwd = MY_PASS,

db = MY_DB)

After the connection is set up through MySQL API, the normal SQL queries can be executed with a cursor object. Below is an example of selecting three columns of the data and print them out.

curs = con.cursor()

query = “select datetime, mac_hashed, category from Main”

curs.execute(query)

results = curs.fetchall()

for row in results:
    print row[0], row[1], row[2]

Besides, PHP scripts are involved aiming at parsing the XML files on YouTube site and retrieving the video information from website. With the input of video ID, the method ‘simplexml_load_file’ is applied to read the feed into simplexml object [56]. After this the XML file is parsed to collect all the attributes we want which are organized as the interested metadata.

In order to simply manage the tables and efficiently handle data in the data warehouse, MySQL Workbench 5.2 CE is used. It is specifically designed for MySQL database modeling via a user-friendly graphical interface. With the help of short queries we can obtain ample basic information and realize such functions as sorting, counting, filtering, manipulating fields, etc.

3.6.2 Filtering Rules

In Figure 2-3 the basic signaling process when retrieving YouTube video clips is illustrated. Due to the limit storage of the data warehouse, two filtering rules were created in the PacketLogic probe. One is all requests with a uniform resource locator (URL) starting with “youtube.watch”, which dumps the subsequent data in each TCP session when a qualified HTTP GET request is found. The other rule is that the “initial flag” is set so we only record the initial part of the TCP sessions, which in our case is the “youtube.watch” request itself.

3.6.3 Data Analysis

The data analysis follows a pre-processing approach as Figure 3-4 depicts. The PacketLogic probe was configured to save the dumped packet files and store them in pcap format on the data dump server. A Python script was then executed to parse the data in pcap files and extracted information such as timestamp, hashed MAC and IP
address, hashed user agent string and video ID. The information was saved in txt files and uploaded to data warehouse, where all further data handling and analysis are performed.
Chapter 4 Experiments and Analysis

The final results are presented in this chapter followed by the conclusions and future work in Chapter 5.

We begin our results and analysis with basic statistics on each parameter and time distributions in section 4.1. The video clip replays and simultaneous downloads within margins are investigated in section 4.2. Section 4.3 describes the network gain and terminal gain in regard to categories as well as contributions of categories to the network gain. Section 4.4 discusses the network gain combined with the video duration. The problems that we encountered and the indispensable hypothesis are listed in section 4.5. Finally a summary in section 4.6 briefly shows the main results that contribute to the motivation of the whole project.

4.1 Basic YouTube Statistics Characteristics

In this typical Swedish municipal network, all YouTube traffic generated by over 2000 households from 9 June to 5 July 2012 is analyzed and the basic characteristics in the dataset are investigated in this section. The whole experimental time lasts for 26 days which contains 3 full weeks. Week 1 is the dates from 11 June to 17 June and Week 2 stands for the dates from 18 June to 24 June. Week 3 starts on 25 June and end on 1 July.

4.1.1 User Type Distribution

In the municipal network, due to the limits of capturing mobile traffic, quite a lot of YouTube traffic generated by mobile users is not recorded. Figure 4-1 presents the user type distribution where the YouTube traffic generated by PC users takes up 98.4% of the total traffic. In our experiments, we focus much on PC user data.
4.1.2 Category Distribution

With the metadata collected from YouTube API, we categorize all YouTube traffic in the same way as defined by YouTube as Figure 4-2 shows. From the pie chart, the music traffic takes the leads accounting for 37.3% of the total traffic, followed by the entertainment traffic (13.7%). These two categories take up half of the entire traffic. The people traffic and Film&Animation traffic consist of 8.1% and 7.9% respectively. The games traffic and comedy traffic make up 7.2% and 6.0%.
4.1.3 Video Duration Distribution

From Figure 4-3, the videos with duration between 3 and 4 minutes are most viewed in the network, accounting for over 17% of all video clips. Over half of watched videos have the duration falling in the range [1, 5] minutes.

![Figure 4-3 Histogram of Video Duration](image)

One phenomenon that needs to be pointed out is that over 10% of video clips have a total length longer than 15 minutes. It is largely due to the cancellation of the 15-minutes video uploading limit [57]. Nowadays users can upload much longer videos at will with only a limit of 20 GB in file size [58].

4.1.4 Video Duration versus Video Category

![Figure 4-4 Video duration in regard to categories](image)
From section 4.1.3, we want to further investigate how the video durations distribute in regard to categories. Figure 4-4 reveals that 97% of the Music videos viewed in the network tend to have a short duration (shorter than 15 minutes) while Games and Film&Animation videos have relative longer durations, with 20% and 27% longer than 15 minutes.

### 4.1.5 User Preferences

![Cdf plot of view counts](image1)

![Pdf plot of view counts](image2)

**Figure 4-7 View counts**

Figure 4-7 and Figure 4-8 demonstrates the cdf - pdf curve of view counts and favorite counts respectively. Half of all video clips viewed in the network have a global view count smaller than 0.15 million times. Only about 10% of total video clips have been liked over 10,000 times and more than 60% of video clips have a favorite count less than 500 times.

Figure 4-9 shows that the average rating is 4.8 as well as only 10% of videos get a rating lower than 4. For all sets of videos that have been observed, the average rating remains above 4.5 with very small variation. As YouTube gets tremendous video library, users tend to look for interesting videos through filtering with some criteria such as the most viewed, the top rated, the latest uploaded, etc. Thus subscribes in our network are much likely to watch videos with a rather high rating.

Our analysis indicates that the two parameters, view count and favorite count, are not well complied in our network. Therefore, to make them more transparent and retrieve the differences between the global site and our network site, some deeper investigations are carried out in the following section.
First of all the global popularity and local popularity are defined as below:

**Global Popular Definition:** If the view count of a video is higher than 1 million times, we consider it as global popular.

**Local Popular Definition:** A video that is requested more than 2.5 times per day.

Besides, we define heavy user and light user with the following standard:

**Heavy user:** a user watching YouTube videos larger or equal to 12 times per day.

**Medium user:** a user with a count of watching YouTube videos between once and 12 times per day.
**Light user:** a user who watches YouTube videos less than once/day on average.

Figure 4-10 compares the percentage of the global popular requests and global popular videos among all requests and all video clips based on different time-scale datasets. It is quite straightforward to see that around 22% of all requests towards YouTube sites are demanding for global popular video download while global popular videos consists of 16% of all video clips watched in the network.

![Figure 4-10 Fractions of global popular requests and videos](image)

Based on the above results, no strong correlation between local popularity and global popularity exists in the network. People tend to choose the contents with a strong dependence of individual interests. This suggests enabling terminal caching for each end user could be a feasible solution to reduce backhaul traffic rather than cache in the network border.

Thus we continue digging into how the amount of traffic for downloading local popular videos looks like. To illustrate it for each end user during the whole experimental time, the requests for popular videos are searched with the total amount of requests to get the end users who watched popular videos. Then the percentage of the requests for popular video clips is calculated based on each end user as below figure shows.

It is obvious to draw the conclusion that the requests for local popular videos that a majority of end users consume are lower than 20% of the total requests. About 15% of end users consume over half of their traffic in watching local popular videos and only three out of them are considered as heavy user. These heavy users contributing much to the traffic of local popular videos attract our significant attentions and will be specifically tracked in future.
4.1.7 Time Distribution in Categories

Figure 4-12 depicts the trends of YouTube traffic in categories with a 24-hours distribution. From the plot, the traffic of each category reach its peak value (about 28%) during clock 19:00 to 21:00 and the valley is hit from clock 5:00 to 7:00, consisting of 3.5% of all traffic for each category. We notice that all categories have the similar time distribution trends of YouTube traffic and over 50% of all traffic occurs between 17:00 and 23:00.
Figure 4-13 reveals the date distribution of the traffic for each category during the whole experimental time. Generally it sums up a high YouTube traffic flow during weekends (which in our analysis is 9, 10, 16, 17, 23, 24, 30 June and 1 July), especially on 24 Jun, which is the mid-summer in Sweden. A high music video streaming traffic is captured on this special day.

4.2 Replays and Simultaneous Downloads

We define a YouTube request as a replay if the video this request aims to retrieve has been downloaded before in the network. And therefore the corresponding video clip is considered as the video played more than once.

During the whole experimental time, approximately 33.3% of the total traffic in our network is replay traffic. The video clips played more than once accounts for 21% of the total video clips that have been watched. So we investigate the replay traffic in this section.

4.2.1 Time Distributions of Replays

Figure 4-14 and Figure 4-15 illustrate the proportion of replay occurrences in regard with 24 hours for weekly and daily respectively. From the weekly chart we conclude that most replays happen during clock 16:00 to clock 22:00 and from 2:00 to 8:00 there are less than 6% of the total replays. Week1 has a relatively high video
replay rate during 18:00 to 20:00, reaching almost 18.5% of the total replays. It might be due to some videos that suddenly get popular or certain users keep consuming a large amount of traffic on one video during that time.

Based on Figure 4-15, Friday wins its highest replay rate from 18:00 to 22:00 as well as Sunday. Saturday postpones its peak value to 22:00-24:00 for that higher chance happens for replays with more users relax with YouTube at that moment. All weekdays enjoy the similar tendency of replay rates.
4.2.2 Simultaneous Downloads in regard to Time Margin

Figure 4-16 shows the percentage of simultaneous downloads among all replays for each YouTube video. The X-axis represents the time margin in seconds between each two adjacent downloads for the same video in our network. It is a key factor to decide the network cache time-to-live (CTTL). The network CTTL is the life cycle of a piece of cache stored usually in the network border before being deleted or re-cached [59].

From Figure 4-16, only 9% of the two adjacent replays for the same video clip occur within 1 minute. Over 70% of the all adjacent replays occur within 50 hours. To save at least 50% of the total replay traffic, a CTTL of 16.4 hours is required.

However when analyzing the tangent trend of Figure 4-16, we can find that the critical point falls when the time margin equals around 20000 seconds (5.55 hours), after which the increase slows down to a constant. Therefore if the cost is great concern and it is in direct proportion to prolonging the CTTL, the optimal CTTL should be set to 5.55 hours, which can reduce about 42% of the total YouTube video replay traffic in the network.

4.3 Network Gain/Terminal Gain with Category

In our investigation, the cacheability is studied in the following three forms: network gain (NwG), terminal gain (TmG) and net gain (NG). In this sub chapter we focus on the cacheability in regard to video category as well as the contribution to the gain for each category.
In the project, the video object size is not traced due to limited storage memory on the servers. Thus the actual watching time of each request is not taken into consideration since we lack the TCP SYN/ACK and FIN/ACK packets. Matching the TCP sessions to corresponding video clip objects is another issue. In our filtering rules of PacketLogic, the initial flag ensures that only the initial HTTP GET request signaling part for each TCP session is recorded. In this way, the number of HTTP GET requests is used to calculate the following parameters which evaluate the cacheability.

**Network Gain**  Assuming a local proxy cache with an unlimited cache size is set on the network border. When a user requests a video clip, the local proxy cache is looked into. If the video clip is already stored in the cache, the content is then directly fetched from cache and sent back to the user. Otherwise, it is forwarded to the YouTube server to retrieve the video content. Once the video content is sent back to the network border, the local proxy cache stores it.

The network gain can be obtained from a local proxy cache and is calculated as:

\[
Nwg = \frac{\text{Total number of requests} - \text{Total number of distinct video clips}}{\text{Total number of requests}}
\]

**Terminal Gain**  If we assume that each end device has an unlimited cache size, all video clips that have been requested by this end node are cached in the terminal. It eliminates the delay of retrieving a video from either a network proxy cache or YouTube server. The terminal gain is defined as:

\[
Tmg = \frac{\text{Total number of requests} - \sum_{\text{for each end user}} \text{Total number of distinct video clips}}{\text{Total number of requests}}
\]

A high terminal gain of the network means each end device requests the same content a large amount of times.

**Net Gain**  Net gain is an indicator of how much we benefit from the replay traffic of a single video clip from multiple users, eliminating the gain of all replays from the same end user. It is a parameter to be analyzed together with the network gain to see if different users request similar video content. If \( N \) is the total number of distinct video clips, then the net gain is defined as:

\[
NG = \frac{\left(\sum_{i=1}^{N} \text{Total number of end users who request video } i\right) - N}{\text{Total number of requests}}
\]

The net gain calculated in our experiments is based on the YouTube traffic in various video categories. For YouTube traffic in each category, the net gain trend is compared with network gain curve to dig out whether the gain is contributed mainly by the replays from the same user or the replays from different users.
4.3.1 Time distribution of network gain in categories

Figure 4-17 and 4-18 show the total network gain of different categories in the network as well as the weekly network gain. In figure 26, traffic of Music (42.94%) and Film&Animation (36.26%) categories have remarkable replay rates compared to that of the total traffic (33.3%).

![Overall network gains in regard to categories](image1)

![Weekly network gains in regard to categories](image2)

From Figure 4-18, the overall network gain in each category drops due to the smaller weekly dataset compared to that of the whole month in Figure 4-17. However,
Music traffic (32%) and Film&Animation traffic (25%) still enjoy a high hit ratio in contrast to that of the total traffic. Some burst network gain occurs in the first week for People&BLog traffic and News traffic due to the reasons like one or several videos about a web-born celebrity or news suddenly become an overnight Internet sensation among certain end users.

The below plot illustrates the network gain tendency during the whole experimental time for the top three categories which have the highest traffic volume. Some special dates give remarkable network gains and these dates are all weekends. The Music traffic arrives over 30% on 12 June due to perhaps some video contents suddenly become famous among one or several heavy users in the network. On 23 June (Saturday) all three categories get their peak network gain and we are all aware of how special that day is for local residents.

![Network Gain Trend](image)

**Figure 4-19 Network Gain Trend**

### 4.3.2 Time distribution of terminal gain in categories

Figure 4-20 and 4-21 illustrate the terminal gains in categories. Seen from Figure 28, the overall terminal gain reaches up to 21%. The Music traffic has a topmost terminal gain over 29%. Thus each end user in our network tends to re-watch the same music video clip that he has downloaded before. For Film&Animation traffic, the terminal gain is not that outstanding as its network gain. It obtains almost the same value as the overall terminal gain has. Games and Sports traffic get the lowest terminal gain, having a value of 11.7% and 10.5% respectively. We conclude that each user in the network does not prefer watching the same game video or a sport event.
In the weekly terminal gain chart depicted in Figure 4-21, the Music traffic still enjoys the topmost (24% on average) despite of the decrease of the value compared to that of overall music traffic. It follows by Film&Animation traffic, which remains constant at around 17% during these three weeks. The other categories show a similar pattern as their overall terminal gain.

It is not hard to notice that People&Blog traffic ranges from 20.5% in week 1 to 17.6% in week 2 to 14.2% in week 3. The obvious fluctuation indicates that terminal gains of People&Blog traffic present a random distribution based on independent user interests. The same random distribution applies to Games traffic and News traffic as well.
4.3.3 Category Contribution to Network Gain

In the previous section we analyze the network gain trends with time distributions. However, even though the network gains in some categories are high enough to draw enough interest, the total share of traffic volume in these categories might take up a small part compared to other categories where large amount of traffic flow is consumed despite of a relatively lower hit ratio.

Therefore, in this section we focus on each category’s contributions to the network gain, trying to retrieve the focus domain where more work should be carried out in future. Related to the proportion of traffic flow in each category as Figure 4-2 reveals, a weighted factor is added to each category. In this way, the weight-added network gain is calculated combining the original network gain and the traffic volume weight.

The weight-added network gain is shown as below in Table 4-1:

<table>
<thead>
<tr>
<th>Category</th>
<th>Time</th>
<th>All data</th>
<th>First half</th>
<th>Second half</th>
<th>Week1</th>
<th>Week2</th>
<th>Week3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music</td>
<td>16.0166%</td>
<td>14.051%</td>
<td>13.6593%</td>
<td>12.227%</td>
<td>12.369%</td>
<td>11.11%</td>
<td></td>
</tr>
<tr>
<td>Entertainment</td>
<td>4.05246%</td>
<td>3.414%</td>
<td>3.4935%</td>
<td>2.992%</td>
<td>3.29%</td>
<td>2.751%</td>
<td></td>
</tr>
<tr>
<td>People&amp;Blogs</td>
<td>2.40732%</td>
<td>2.1084%</td>
<td>2.1141%</td>
<td>2.086%</td>
<td>1.87%</td>
<td>1.568%</td>
<td></td>
</tr>
</tbody>
</table>
Table 4-1 Weight-added network gain

<table>
<thead>
<tr>
<th>Category</th>
<th>Added Network Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Film&amp;Animation</td>
<td>2.86454%</td>
</tr>
<tr>
<td>Comedy</td>
<td>1.7214%</td>
</tr>
<tr>
<td>Games</td>
<td>1.6776%</td>
</tr>
<tr>
<td>Sports</td>
<td>0.6464%</td>
</tr>
<tr>
<td>News</td>
<td>0.5203%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Added Network Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Film&amp;Animation</td>
<td>2.4861%</td>
</tr>
<tr>
<td>Comedy</td>
<td>1.3422%</td>
</tr>
<tr>
<td>Games</td>
<td>1.3255%</td>
</tr>
<tr>
<td>Sports</td>
<td>0.56296%</td>
</tr>
<tr>
<td>News</td>
<td>0.5028%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Added Network Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Film&amp;Animation</td>
<td>2.43478%</td>
</tr>
<tr>
<td>Comedy</td>
<td>1.4856%</td>
</tr>
<tr>
<td>Games</td>
<td>1.60128%</td>
</tr>
<tr>
<td>Sports</td>
<td>0.53227%</td>
</tr>
<tr>
<td>News</td>
<td>0.42048%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Added Network Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Film&amp;Animation</td>
<td>2.16073%</td>
</tr>
<tr>
<td>Comedy</td>
<td>1.125%</td>
</tr>
<tr>
<td>Games</td>
<td>1.107%</td>
</tr>
<tr>
<td>Sports</td>
<td>0.4898%</td>
</tr>
<tr>
<td>News</td>
<td>0.5273%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Added Network Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Film&amp;Animation</td>
<td>2%</td>
</tr>
<tr>
<td>Comedy</td>
<td>1.292%</td>
</tr>
<tr>
<td>Games</td>
<td>1.128%</td>
</tr>
<tr>
<td>Sports</td>
<td>0.439%</td>
</tr>
<tr>
<td>News</td>
<td>0.389%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Added Network Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Film&amp;Animation</td>
<td>1.97%</td>
</tr>
<tr>
<td>Comedy</td>
<td>1.131%</td>
</tr>
<tr>
<td>Games</td>
<td>1.4422%</td>
</tr>
<tr>
<td>Sports</td>
<td>0.4576%</td>
</tr>
<tr>
<td>News</td>
<td>0.36%</td>
</tr>
</tbody>
</table>

Figure 4-22 Category contributions to the overall network gain

Figure 4-22 gets the statistics in table 4-1 and displays the contribution rates of each category to the overall network gain. It is quite transparent to see that the Music traffic contributes nearly 54% to the overall network gain, which takes the leading
role to decide how the network gain looks like. Entertainment and Film&Animation rank the second and third places, contributing 13.6% and 9.5% to the overall network gain.

The above results indicate that some categories like Film&Animation have a low contribution to the total network gain even though it enjoys the second highest network gain. However, categories like Music obtain both ideal category network gain and category contribution. This arises our interesting in Music traffic, which is further analyzed in section 4.3.5.

### 4.3.4 Network Gain & Net Gain in Categories

A study of network gain combined with net gain is included in this section. The goal is to understand the constitution of the network gain in each category. In another word, the gap between the network gain and the net gain tells whether there are many users requesting the same YouTube video content or a small number of heavy users who keep replaying the same video clip. The smaller the gap is, the more different users there are who demand the same video content rather than certain heavy user keep watching the same video clip.

Figure 4-23 is a comparison of the gap within all categories. In all categories column, the network gain is around twice higher than its net gain. So a multiplying factor 2 is applied in the following each category. Apparently, the net gain of Music traffic is 19.5%, which is smaller than half of its network gain (43.1%). The same case happens to People&Blogs (NwG=29.8%, NG=12.1%) and News (NwG=21.7%, NG=8.8%). These 3 categories contain more traffic of which single heavy user consumes large amount of replays.

![Figure 4-23 Network gain and Net gain in regard to categories](image-url)
On the other side, the net gain of Film&Animation traffic reaches as high as 19.5%, exceeding half of the network gain (36.2%). Since Film&Animation traffic has an optimal hit ratio, this result declares that many families in our network share the common interests towards Film and animations. Besides, Comedy, Games and Sports also have a relatively high net gain, indicating that multiple users tend to share the same appetite towards topics like funny performances, games and sports.

Based on the previous overall network gain contribution results, we pick up the dataset of the topmost three categories: Music, Entertainment and Film&Animation and get the statistics of network gain (NwG) and net gain (NG) in weekdays as Figure 4-24 illustrates.

![Figure 4-24 Network Gain & Net Gain weekday distribution](image)

On average, the network gain of Music traffic goes flat except for Tuesday, when it reaches the peak value of 27%. Observing that the net gain of Music traffic stays low on Tuesday as other weekdays, we consider that the reason is related to a high proportion of replays for a suddenly popular video clip consumed by certain heavy users.

Similar to Music traffic, Entertainment traffic stays steadily on both network gain (16.4%) and net gain (5.8%). However, Film&Animation traffic has a rather high network gain on Fridays and Saturdays compared to other weekdays. We conclude that it is most probably because of the replays of certain local popular animations requested from work-free parents with their after-school children as well as the coming weekends on the next day. From the net gain charts, the Film&Animation traffic gets its peak on Fridays and Sundays. Thus, most replays of Film&Animation traffic are contributed by different users requesting the same video content on these two days.
4.3.5 Network Gain of Music Traffic – Further Analysis

From all above results, Music traffic enjoys the highest network gain compared to other categories with a relatively low net gain in regard to its network gain. In addition, it has over 50% percent of the contribution to the overall network gain. Although it goes steady in Figure 4-24 during 7 days per week, the network gain could be interesting to investigate for 24 hours per day.

We can easily find from Figure 4-25 that the peak value of Music network gain happens around 18.00-20.00 in the four different datasets, especially the value reaches 29% during week 1. The total network gain during that time gets 26%, nearly twice of the valley value, which appears during clock 0.00 to 4.00.

It is also appealing to notice that some unexpected explosions exist, for example during 4.00 to 6.00 the network gain of week 2 leaps up to 22.9% and during week 3 the highest network gain occurs in the [8.00-10.00] time interval.

![Figure 4-25 Network gain Hour Distribution of all Music traffic](image)

4.4 Network Gain with Video Duration

Section 4.3 lays the concentration on the network cacheability in regard to the video clips belonging to categories. In this section, the cacheability is analyzed taking the video durations into account.

The assumption is a prerequisite that the local proxy cache is aware of the duration of each video clip that is requested by local end users and it has the
capability of deciding to cache videos of certain duration ranges. Since the video duration can be obtained when retrieving the html page, it will not be difficult for a proxy cache to detect the duration and take decision based on certain metadata of the video contents in future. Therefore, if it could make the caching strategic decision based on the video category and video duration, a high optimization will be realized according to the results both for smooth video watching and the backbone traffic volume offloading.

4.4.1 Network gain with video durations

Figure 4-26 reveals the network gain in regards to the video duration during the entire experimental time and each week. From it we derive a quite optimal network gain of the videos whose durations fall between [3, 4] and [4, 5] minutes, which reaches as high as 42.4% and 40.6% respectively. For the weekly datasets, the similar curves are displayed and videos with duration between 3 and 5 minutes have 1.9 times higher network gain than the lowest value.

The network gain of video clips whose durations are longer than 30 minutes rises up to 33.6%. It arouses our curiosity since the long video clips are more likely to be watched several times by users who share the common interests. However, it is another issue which part of the content patterns and how large size of the video clips should be stored in the network to obtain a highly efficient cacheability. This could lead to a hot topic in the future work.

4.4.2 Network gain of Categories with video durations

In this section the video duration and category are both considered. In the Figure 4-26, Music and Film&Animation traffic generally reach a higher network gain than that of total traffic (the back line in the figure) and other categories present below the
total traffic, which complies with the above results. In general the network gain of the total traffic shares the same trend as that of Music traffic. This certifies the above results that Music traffic contributes over 50% to the overall network traffic.

For all videos with duration between 3 and 5 minutes, Music traffic obtains its peak value around 47% while Film&Animation traffic gets 35%. But Film&Animation reaches its peak value (43%) for the videos whose duration falls into [5, 6] column. It indicates that the Film&Animation videos that are locally popular tend to have longer durations than that of locally popular Music video clips.

Entertainment and People&Blog remain close trends as all data traffic does. The network gains of Games, Sports and News traffic are lower than that of all data traffic partially because of the small amount of traffic and independent user interests.

![Figure 4-26: Network gain of video traffic of categories with different durations](image)

**4.5 Problem and Hypothesis**

Due to the continuous expand of YouTube streaming traffic according to the prior work, an efficient measurement framework is of paramount importance to find ways of reducing the cost and the resource consumption. The amount of YouTube traffic flows that can be freed up by introducing a smart cache strategy who knows video information is measured. YouTube video metadata is dig into to get the interesting parameters. The locality within regions shows if a network cache is necessary when...
we know the category of the YouTube video or the clip duration. If so, how large the cache is and how long the content object is stored in the cache will be further investigated. The popularity prediction is also a concerning problem to dig out in the project. Last but not the least, the multiple meta-data of video contents watched by each end user is considered to categorize them. Through this, high pertinence on the content demand patterns of the dataset is realized.

Measuring and analyzing YouTube traffic in a Swedish commercial municipal network encounters some obstacles. Hypothesis should be made to adjust experiments given the validity and authenticity of the data. The results obtained are based on the final experiment to which all the following hypotheses have applied.

- The statistics is based on the user generated contents collected in the municipal networks Acreo has access to, which has a characteristic of region and network access mode. For example, videos about regional news are only popular among certain areas and most of the subscribers are FTTH households.

- Since the identification of end users is decided by the gateway Mac address plus the user agent string, two end devices that use the same web browser under the same Mac address fail to be distinguished. In such case, hypothesis is set that each family is considered to own only one end device with such user agent string. Therefore, the actual number of end devices is underestimated.

- The definition of video categories is given by YouTube. It now has totally 15 categories: Autos&Vehicles, Comedy, Education, Entertainment, Film&Animation, Gaming, Howto&Style, Music, News, Nonprofits, People&BLog, Pets&Animals, Scientific&Technology, Sports, Travel&Events. Whenever a video is uploaded, it is placed into categories according to its content in order to facilitate viewers to find their interested videos. However, some video contents are critical towards several categories and putting it in only one category seems a little improper. Since the category of each video can be changed by the publisher, the categories of around 5% videos are considered not accurate in all my experiments. Certain error is allowed within the bounds.

- The YouTube Application Programming Interface (YouTube API) [50] provided by Google Developers, enables applications to execute normal functions on YouTube website and to retrieve standard video feeds, video information and comments. In the project, I keep sending the GET request towards YouTube API to retrieve the metadata of each video. Since the number of requests received by YouTube API is quite large, the access from my workstation is blocked after a certain time because of the defense mechanism against DoS attack. Therefore, I have to wait a period of time and then manually restart sending the requests.

- Some metadata collected by YouTube is not up-to-date and it always takes time
before these statistical variables get updated, like the view count, the favorite count, the rating, etc. For certain outdated videos, their statistics might be far from accuracy.

- The increasing amount of encrypted data adds the challenge to the traffic analysis. The sensitive subscribers’ data is protected by law and is not supposed to be eavesdropped or tampered. No raw data is allowed to leave the operator network before it is encrypted and hashed. Besides, some P2P end users attempt to encrypt the file-sharing traffic as they don’t want it to be identified by ISPs. The traffic could also be transmitted through a HTTPS or SSH tunnel. Subject to these limits, several parameters of interests fail to be investigated.

- The privacy issue comes up since subscribers don not want to leak their user data to others through ISP’s network. However, it happens when some ISPs agree to provide certain information of subscribers to third parties. In this case, the sensitive user privacy should be carefully erased but the meaningful data is collected for better analysis. For massive data processing, parts of the interested data might be missing or get erased due to subscriber privacy policies.

### 4.6 Summary

To sum up the above results, the following 5 main aspects are listed derived from the analysis concerning different traffic content patterns and its caching gains.

1) Music traffic gets an ideal network gain (42.94%) together with a high contribution to the overall network gain (54%). Film&Animation traffic wins the second highest network gain, which reaches 36.26% but the volume is not large enough to drive the overall network gain compared to Music traffic and Entertainment traffic.

2) Music traffic enjoys a remarkable terminal gain of 29% but a low net gain (19.5%), indicating that its high network gain are more due to some single heavy users generating large traffic of the same video contents. In contrast, Film&Animation traffic has a low value of terminal gain as the total traffic has, which is 21%. But its net gain reaches as high as 19.5%, the same figure as Music traffic has, especially on Friday. This reveals that the network gain is mostly contributed by multiple end users retrieving the same video clip. The number of multiple end users gets the peak value on Friday due to the early off-school for kids.
3) The network gain of all traffic requesting video clips with a length between 3 and 5 minutes reaches ideally over 40%.

4) More than half of the replays occur during 16.00-24.00 and peak hours always appear on average from 18.00 to 22.00.

5) Only around 16% of the videos are global popular and very few heavy users tend to be local popular video viewers, showing some region limitations and independent user interests.
Chapter 5 Conclusions

5.1 Conclusions from Backgrounds

The background of the thesis project is studied and concepts of all related fields are described after being sorted. The fundamental strategy of QoE and cacheability is involved with respect to YouTube policies. The previous researches worldwide on content patterns and user behaviors of YouTube media streaming traffic are consulted. This section has a high generality as an input of knowledge of related researches as well as a prerequisite of future work on YouTube content pattern investigation and cacheability with video metadata.

5.2 Conclusions from Goals

This thesis project aims at analyzing the content demand patterns of YouTube video streaming traffic as well as the cacheability reflected from the statistics during the experimental time of traffic tracing and packet monitoring for a typical Swedish municipal network. The goal is well fulfilled as expected. Interesting results have been obtained from a new and efficient approach of traffic patterns monitoring and massive data processing. More content demand patterns with a longer experimental time are supposed to be involved at the first beginning. However, because of the physical resource problem and time limit, they are put in the future work.

5.3 Conclusions from Methodology

The approaches deployed in this project can be widely used in other similar investigations and the Python/PHP scripting has been optimized for customizing other kinds of traffic monitoring and data parsing. For instance, analyzing the traffic of SVT [60], known as the most popular Swedish public TV service, can also utilize the similar methodology as we used in the project. The corresponding rules of SVT video
streaming flows can be analyzed and summarized before applying a set of traffic dump rules. With all collected data, the comparable data analysis can be performed as well as the way to achieve interesting parameters and user behaviors.

Moreover, the user identifier on which we base the subscribers in our network is a hashed 32-bytes string of the household MAC address combined with corresponding user agent string. This method enables the consistent identification of unique end devices and meanwhile protects users’ privacy. The hashed end user ID string acts well during the project and is more accurate than using an IP address or a MAC address in the previous work since the same IP address could be dynamically allocated to multiple end devices for a certain amount of lease time and the MAC address is not precise enough to track all end device due to various laptops, tablets and smart phones in one household.

The methodology is benefited from the previous work and will provide a feasible universal approach for the future investigations. The conclusions obtained are also the fruits of a continual work inherited from the previous experiments in Acreo and are going to lead the future research directions to some degree. This project demonstrated massive data collection and analysis of YouTube video traffic regarding the cacheability and user behaviors by classifying the video contents in category and duration.

5.4 Conclusions from Experiment and Analysis

The measurement framework that is constructed in this master thesis has a good compatibility to other applications other than YouTube. Similar investigations on SVT streaming traffic has been performed under the same network architecture and it turned out expected results with a low complexity. The framework is generalized to applications but it remains to see if it applies to multiple network environments.

From the basic statistics of content patterns, the Music traffic and Entertainment traffic consist of over half of total traffic volume. When it comes to the video duration, over half of total video clips have the length within 5 minutes. Since only around 16% of the videos are global popular and 22% of the total traffic is consumed for global popular video clips, no strong correlation between the local popularity and the global popularity exists in our network and users tend to choose the contents with a strong dependence of individual interests. Furthermore, the heavy users contribute little to the local popularity. The time distribution of the video streaming traffic in various categories provides us a clear view that weekends and red days are the dates with the highest traffic volume for each category. The peak hours occur on average from 18.00 to 22.00 and valley load hours appear during 3.00 to 5.00.

The time distribution of the replay traffic is analyzed, which turns out a similar
An infinite size of the proxy cache was assumed throughout the experiment. The cacheability was evaluated based upon the network gain, terminal gain and net gain. The network gains of traffic in different categories are thoroughly studied to get the possible common interests. Our results reveal that Music traffic enjoys an outstanding network gain (43%) together with a remarkable contribution (54%) to the overall network gain. Entertainment traffic takes the second place in the share of the whole traffic volume but no particular cacheability appears for us to get benefits from its hit ratios. The other category whose network gain reaches as high as expected is Film&Animation (36.3%). However, the contribution to the overall network gain is quite weak compared to that of Music traffic.

As for terminal gain, Music traffic gets a remarkable terminal gain (29.1%). Film&Animation and People&Blog traffic have the similar height of the terminal gain as that of the overall terminal gain, which is about 20%.

The net gain results explore further how the network gain is contributed. The low net gain in contrast with the highly optimal network gain in Music traffic tells that more replays for Music video clips are demanded by one or several heavy users who download the same video contents several times. On the opposite, Film&Animation traffic comes with a high net gain (20%) indicates that multiple users in our network share some common interests towards film videos and animations, such as the newest film trailers and Swedish popular children programs.

Assuming the proxy cache is aware of the video duration whenever a video clip is downloading from YouTube site into our network. Then our results show that the network gain of all YouTube traffic for requesting video clips with a length between 3 and 5 minutes reaches up to more than 40%. Besides, the traffic for the video duration larger than 30 minutes has a network gain of almost 35%, showing partly certain common user behaviors on the long videos. When we combine the duration together with the video category, the Music traffic tends to have the similar curve as the overall traffic has, whose peak values are both in the duration interval [3, 5] minutes. However, the Film&Animation traffic obtains its top value of the video clips with duration between 5 and 6 minutes. Therefore, it would be more efficient to enable the network caching given the knowledge of video duration and category on the proxy server where caches are stored. For Film&Animation traffic, a proxy server cache set
in the network border (could be a base station or a backbone router) is a feasible solution while Music traffic can be deployed with a terminal-caching strategy in order to reduce the backhaul link traffic of the same contents retrieved from certain heavy users.

### 5.5 Personal contributions

This section lists the personal contributions of this thesis project to the future work and other researches in similar domains.

The main contributions can be summarized into two parts: the new measurement framework to connect cacheability with video clip metadata and the confirmation of some universal knowledge verified by these experiment results.

The new measurement framework based on the network environment can be easily setup and reused in future researches. Parameters are configurable to create interest-driven scenarios. It can also be adjusted to other online media services such as SVT without complicated changes.

Besides, from the results of the experiments running on this framework, some universal knowledge that has been taken as granted is confirmed by the figures in the Swedish municipal network. Certain content patterns and cacheability laws are retrieved with concrete statistics to provide evident inputs for future work.
Chapter 6 Future Work

In this thesis project the content demand patterns of YouTube traffic is studied as well as the cacheability in regard to the video category and video duration, under the assumption that the network cache of the proxy server and the terminal cache of the end devices have unlimited sizes. However, to design a more intelligent caching mechanism, such assumptions should be re-defined and the up-limit value should be set to constrain the video objects that have a huge size. Moreover, during the whole experimental time (almost one month) we never clear any content in the cache. Thus all the video clips that are new to the network are stored in cache since then till the end. In another word, the cache life-cycle is set to infinity. Future work can improve this point by proposing a practical model to predict the cache life-cycle. Some related investigations can also help, such as tracking the newly watched video clips in the network to find out their popularity grows and drops. After gathering all popular durations of these videos, some inputs can be fetched for building such cache life-cycle model. Furthermore, upon the cache is full, decisions should be made to empty some parts of cache for the new content depending on the principals that have been defined.

Another part that can be improved is a more fully-covered capturing rule. Since in our experiments lots of traffic generated from smart phones is not able to collect because no certain signaling packets are sent out when mobile user fetches embedded YouTube videos in some pages or on apps. Further analysis of why the missing packets happen in these situations can be put forward.

This project focuses as well on the video duration, which is one of the video metadata. However, researches are highly eager to find out what the actual playback time is for each video clip watched by users. With the actual playback time, the object sizes can be roughly calculated which benefits much the cache size estimations. The next phase could be then to analyze the TCP sessions for each HTTP request towards a certain video clip. However the tricky part is how to map the TCP FIN/ACK session to the corresponding video playback session as the same video ID could generate several related TCP sessions. An alternative solution of this could be linking the video playback session and the TCP FIN/ACK session via the port number.
References


[50] YouTube API v2.0 Documentation. Last viewed July 2012. [Online] Available at: https://developers.google.com/youtube/2.0/developers_guide_protocol


