Discovering the working mechanisms of YouTube using network diagnostic tools

1nd B. Sc. Fabian Hörst
Ruhr-University Bochum/Aalto-University
Bochum, Germany and Helsinki, Finland
fabian.horst@aalto.fi and fabian.hoerst@rub.de

***Abstract*—The following article deals with the working mechanisms of YouTube, which is discovered by using network diagnostic tools like WIRESHARK [1], TRACEROUTE [2] and the developer toolbox of the FIREFOX-browser. The aim is to find out the network architecture of YouTube and the workflow of video search and delivery. Moreover, the structure of the "content delivery network" (abbr. CDN) is evaluated by multiple test cases. The article is written for the course "basic principles in networking", performed at Aalto University, based in Espoo, Finland.**

I. INTRODUCTION

YouTube, founded in 2005, with the corresponding website "www.youtube.com", is one of the most accessed web-services, to be exact the second most visited web-service after the search engine "www.google.com". [3] YouTube is the biggest provider for video streaming and has 2 billion logged-in monthly users. The platform ranks on the second place in global internet traffic. [4] Thus, YouTube is a very popular and powerful web-service. In this article the infrastructure of YouTube is discovered by using multiple network diagnostic tools. The main focus lays on finding the used protocols, entities involved and how the user can interact with the website. The workflow of video search and the delivery of content is described. Because of the large network traffic caused by YouTube, the web-service uses content delivery networks for delivering the requested data. The fact, that YouTube is a subcompany of Google LLC is important regarding to the structure of the CDN and is considered later in this article (section III-B).

Structure. To understand the working mechanisms of YouTube, the interaction between the user and the web-services needs to be analyzed. Therefore some test cases and experiments are necessary. In section II, the experimental setup and the used software for the measurements are introduced and explained. The section is divided into two parts, whereby II-A deals with the experimental setup for discovering the general architecture of the whole web-service and II-B deals especially with the setup detecting the structure of the CDN used by YouTube. The measured data and results gained from these experiments are explained in section III. This section is divided into two parts, in which III-A is discussing the general architecture, used protocols and involved entities, the workflow of video search and delivery and III-B evaluates the CDN structure. Further information and used literature is given in the References on page 6 at the end of this article.

II. EXPERIMENTAL SETUP

A. How to figure out the architecture of YouTube

The aims of the experiments are to discover the network architecture, the workflow of video search and video delivery. All the following experiments are carried out on a PC with Ubuntu (version 19.04) as the operating system and Firefox (version 72.0.1) as the web-browser. To get the network architecture, the website of YouTube is accessed and the network traffic is measured with the tools Wireshark and the Firefox developer tools. To get the necessary information, first, the website is accessed. After that, a video search is performed for the video "Hotline Bling" [5]. With this setup, one part of the network architecture and die video search can be discovered. After that, to get information about how the video data is streamed to the user, the video "Hotline Bling" will be viewed and again the network traffic is measured. To get more information about video delivery, the video is played again, but the video quality is changed during watching the video and also stopped after a few minutes. This aim of this is to figure out how the web service behaves if the user is interacting with the web service during video streaming. All these experiments are also repeated with different kinds of videos and also at different times.

For the experimental setup, there are a few things that need to be mentioned. Accessing the YouTube website via the hostname "www.youtube.com" shows that YouTube uses the "Hypertext Transfer Protocol Secure" (short: HTTPS). This can be seen in the address line of the browser, in which the used communication protocol can be recognized. With HTTPS, the data is encrypted with the encryption protocol "Transport Layer Security" (short: TLS), often known as the "Secure socket layer" (short: SSL) [6]. A more detailed description of this is included in the evaluation of the experimental results in section III-A. The use of HTTPS has an impact on the use of network diagnostic tools since the data is only decrypted in the application layer (here: Firefox web browser) using a private key [6]. If the development tool from Firefox is used, the decrypted data can be read and evaluated without any problems. When using Wireshark, however, the data must be decrypted and the private key needs to be known because Wireshark receives only the encrypted packets. Instructions on how to encrypt TLS encrypted data in Wireshark can be found in reference [7]. The procedure in the test environment (Firefox on an Ubuntu distribution) can be summarized as follows:

- Open a terminal and type in the following command: `export SSLKEYLOGFILE=\$HOME/sslkeylog.log`
  This creates a SSLKEYLOGFILE environment variable, in which the TLS key is stored
- In the same terminal, open Firefox with: `firefox`
- Open Wireshark and go to edit → preferences → protocols → TLS and there store the created file sslkeylog.log in the field "(Pre)-Master-Secret log filename"

With these settings, the entire network traffic from YouTube can be analyzed using Wireshark and the Firefox development tool. The tool Traceroute [2] can be used to determine the route via which the data gets from a client to a server or vice versa. The location of an IP address is queried via the API of the web service ipinfo.io.

B. Introduction to CDNs and how the structure is figured out

Introduction to CDNs. Before the experimental setup to discover the CDN structure of YouTube is explained, a brief introduction of CDNs is necessary. A CDN is a distributed server network, in which the servers are located in different geographical locations. These servers are often located on internet exchange points (short: IXPs). A CDN is pulling static content files from the origin server into his distributed network, which is called caching. The static data is delivered to the end-user by the CDN and not by the origin server [8]. It is an optimization of the quality of experience (short: QoE) and the quality of service (short: QoS), whereby these two optimization goals are subject to economic constraints. QoE means subjective user experience, which implies the entire experience of the web services from the subjective point of view of the user, whereas QoS describes neutral quality based on measurable data like round trip time (short: RTT) or bandwidth usage inside the network [9], [10]. As mentioned in I, video streaming causes an extremely high amount of data. Since the number of users is also very high, it is almost impossible to send all streaming requests to a server and to obtain the data from it. That is why YouTube uses a CDN to cache the content in different locations around the world. This provides additional bandwidth, takes the load off a server and delivers the streaming content to the user without needing to transmit the data within large distances. The goal is that long loading times should not occur for the website, the highest possible quality of content is delivered to the end-user and to achieve high reliability for streaming, so that the video can be displayed without any interruptions. All in all, the advantages of a CDN can be summarized as follows:

- Copies of the video data can be stored as close as possible to the user. This results in lower loading times.
- CDNs are scalable, this means the CDN can be extended by adding new servers to it.
- Storing the same data on different servers leads to data redundancy. Compared to a single-server structure, this can reduce the risk of data loss because probably a copy of the data exists on another server.
- If one server has to manage a high data traffic at one moment, a request can be redirected to another server with lower data traffic to improve the QoS for the user.

Experimental setup. To find out the structure of the CDN of YouTube, it is necessary to watch different kinds of videos from different locations at different times. The change of location of the client is necessary to figure out whether the location has an impact on the server, from which data is delivered. The different times for measurements can show what is happening on times with probably high network traffic and low network traffic. The different kind of videos is used to find differences in the delivering CDN server depending on the kind and popularity of the streamed video. To measure all things in a structured way, the following procedure was developed:

1) The location of the client is Helsinki, Finland.
2) Open the Firefox developer tools or Wireshark.
3) One after one watch every video listed below and measure the network traffic of the HTTP-protocol:
   - Despacito [11], the most viewed video on YouTube
   - Au Revoir [12], a popular German music video
   - Skiing Hintertux [13], an unpopular German video
   - Hei rakas [14], a popular Finnish music video
   - Teekkarihymni [15], an medium popular Finnish music video
4) Save the IP-Address of the delivering video data server, seen in the network traffic.
5) Run the shell program given in listing 1. It does a TRACEROUTE call for the saved IP-Address and resolves the server locations over an API of the web-service "ipinfo.io". The program prints out the locations of the servers and nodes passed until the CDN server of YouTube is reached.
6) Restart the procedure at point 1) for the different locations of the client listed below (the location is changed with VPN software): Brisbane (Australia), Bochum (Germany), Chișinău (Moldova)
7) Also repeat the steps 1)-6) for different times, such as morning (probably low network traffic in Europe) and evening (probably high network traffic in Europe).

The locations for the clients were choosen due to the fact they are in different countries and different continents. For selecting the videos, the criteria were popularity and the origin. The criterias can also be seen in table I.

```
# printing out the locations of traceroute
(traceroute -m 60 -n $1 | awk '{print  $2}') > /home
    /fabian/ip.txt
while read variable
do
    if [ $variable != "to" ]
    then
        if [ $variable != "*" ]
        then
            curl ipinfo.io/$variable/city?token=508\
                d4763aa8b9d
        fi
    fi
done < /home/fabian/ip.txt
```

Listing 1. Source code for showing the locations of the servers and IP-network nodes during traceroute lookup.

III. Measurement Results and Evaluation

A. Network Architecture: Entities and their roles, workflow of video search and delivery

Because the network architecture is connected with the ways how the videos are searched and delivered to the user, both aspects are evaluated in this section. At first, the basic structure of the network is inferred. As mentioned above, for this it is assumed, that the video content is delivered from a CDN. The structure of the CDN is not part of this section, rather than from section III-B. The evaluation of the address line of the Firefox web browser shows that YouTube uses HTTPS as the communication protocol. The aim is to ensure encrypted data transmission, with encryption being added as an additional layer in the transport layer of the TCP/IP model. If the website is called up and the network traffic is measured with Wireshark, one can see that the TLS version 1.2 and 1.3 is used as encryption. The website is also transmitted with the TCP protocol on the transport layer. The TCP/IP model for YouTube is shown in figure 1. HTTPS is syntactically identical to HTTP and only the encryption has been added. The standard port for this is port 443/TCP [16]. If a user calls up the website "www.youtube.com", the hostname is resolved using DNS and the user is forwarded to a server belonging to this hostname. Visits to the website at different times and in different places show, that the user is not always forwarded to a server with the same IP address, but the servers change. However, the location of the servers is always Mountain View, US. The server then sends the website with the content to the user. No video data is loaded here, only the website with video recommendations and .img files used as thumbnails.

Search of a video. If the user is starting to type in letters into the search entry box, after every letter a request to the YouTube servers is send and the response is a JavaScript file, containing the suggested videos, which are shown in a drop-down box. Thus, until the user has not click the search button, the website is just loading this content. After clicking the search button, the search keyword is sent to the server with HTTP and the server returns the search results and sends thumbnails of the suggested videos. The style-content of the website is the same and not new loaded. So YouTube has a base website, which remains stable during the search of the video and just the results displayed are changed.
Delivery of a video. If the user clicks on one of the videos, he will be directed to a new website. This is illustrated here with the video "Hotline Bling" [5]. Clicking on the video triggers an HTTP request event, which forwards the client to a new website. A query string at the end of the URL is used, which contains the ID of the video. Each YouTube video has an ID, as do the channels. The structure of a video link is shown as an example in Figure 2. After the request, the YouTube server replies and provides the layout of an HTML website in which the video is embedded. HTML 5 is used, which enables the direct embedding of videos of the MP4, WebM, and Ogg file type, which means that these videos can be played directly in the browser without any additional software. Many other files are loaded from the YouTube servers, including CSS files for the design and JavaScript for other plugins.
The video itself is not loaded from the server hosting the website. Since videos are static data that are not created when the user accesses them, they can be cached in a CDN and delivered to the user. The client receives the address of the CDN server from which it should get the data via an HTTP response. The address of the CDN server has the form: rx- - -sn-y-ixhs.googlevideo.com/videoplayback? where x is a number and y is a character string. Also, a keepalive TCP connection is established with the server from which the video data are obtained. In addition, the client sends data to the YouTube server using the HTTP post method and receives an HTTP response with code 204 [no content], which means that the

Application	HTTP
Transport	SSL/TLS (TLS v1.2)
	TCP
Internet	IP
Network Interface	Ethernet

Figure 1. TCP/IP-model of YouTube

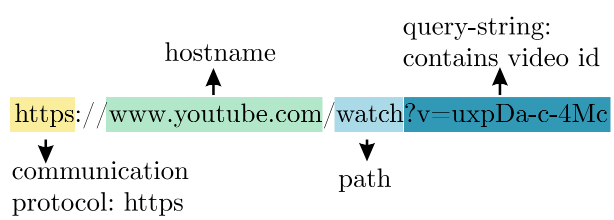

Figure 2. URL structure of the video "Hotline Bling"

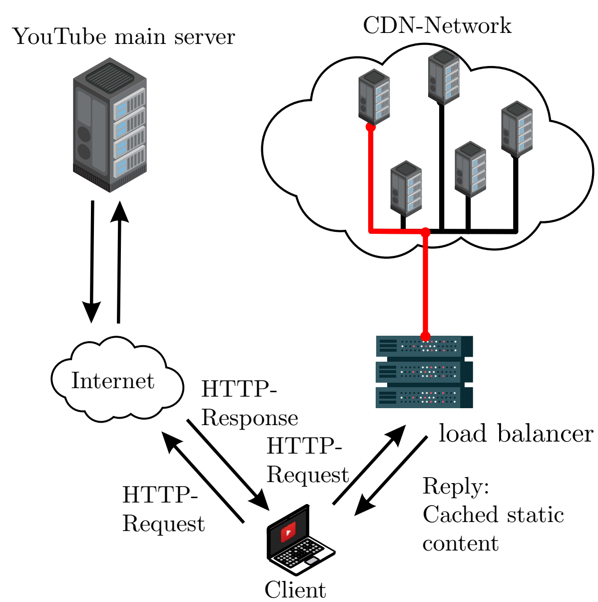

Figure 3. Network structure of YouTube for accessing a video

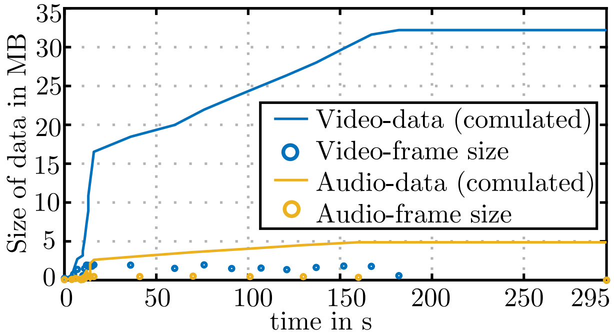

Figure 4. Video data transfer during watching the entire video "Hotline Bling"

server has received the data. The data contain statistics such as the current watch time and probably also information about video quality. The entire network structure is again shown graphically in Figure 3. The load balancer is introduced in section III-B.

For the case described here using the "Hotline Bling" video, the video data is transmitted in the .mp4 format, whereas the audio track is transmitted separately with the .webm file format. But other measurements showed that there are also other file formats possible. The amount of data transferred to the client is shown in figure 4, the data is shown separately for audio and video data. The cumulative amount of data and the size of the individual data packets can be seen, plotted over the playing time of the video. It shows that a lot of data is loaded at the beginning of the video. This can also be referred to as the initial buffering of the video. After that, the amount of data obtained drops and is only called up at largely regular intervals. This can be explained as follows. At the beginning of the video, a lot of data is required to buffer the video on the device. However, the entire video should not be loaded directly, on the one hand not to unnecessarily load the memory of the device, but on the other hand not to cause unnecessary network traffic, since a video may be interrupted during playback or the quality is changed. Thus, only parts of the video are reloaded after the initial buffer phase, whereby the data throughput is almost constant and slightly above the data rate of the video because the complete video was loaded before the end of playback time. If the video is stopped, no network traffic occurs. If the video quality is changed, there is again the HTTP post method raised to inform the servers, but the CDN server is not changing.

B. CDN-Architecture

The results for the experimental setup explained in section II-B are given in the table I. The data was measured on the 16th of February 2020 at 8:00 and at 18:00 o'clock (Timezone: UTC + 2:00 hours). Unfortunately, the time did not affect the selected streaming server and the measured data was the same. Therefore, table I is valid for both times. In figure 5, the locations for the clients, the servers and the route of the streaming data are illustrated in a world map. The route of the data is just estimated because traceroute was not able to resolve many IP-Addresses due to the fact that many network nodes are not responding to it. Therefore the routes can also follow different ways, but in the figure 5 the maximum likelihood route is illustrated based on the IXP nodes and the undersea optic fiber cables shown in reference [17].

Evaluation. With the experimental setup, different content servers of the CDN were located in Helsinki (Finland), Perth (Australia), Frankfurt a. M (Germany), Bucharest (Romania) and Mountain View (United-States). The server in Mountain View is based in the headquarter of Google LLC and is one of the main servers of YouTube and Google. YouTube is using the CDN of its parent company Google LLC. This can be verified by looking up the owner of the servers with the web-service "ipinfo.io". This fact leads to the assumption, that YouTube has not to pay any fees for using the CDN, so the economic constraints are not in the foreground. What is meant with that, is if a company has no own CDN, the company needs to engage a CDN service provided by a CDN provider, which costs the company money for the traffic in their CDN. In this case, cost optimization is very important. As opposed to this, if a company can use their CDN, they have more degrees of freedom to optimize the traffic in the network without thinking about the cost.

As shown in figure 5, for the most popular Video "Despacito" [11], YouTube tries to deliver the content to the user from the nearest geographical server to the client. An exception is Brisbane. A closer Google LLC data server is based in Sydney, but the content is delivered to the client by a server in Perth, which is more far away. The reason for this exception is probably load balance in the network because in Sydney there is an IXP node responsible for all the network traffic between Australia and United-States and therefore the servers in these locations may have high network traffic. Further experiments could also show, that the network traffic depends on the time and therefore the selected content server of a client at one position can change for the same video on different times, like in reference [8]. Not only the load balance in the network but also the content of the video and the popularity is having an impact. For example, the most popular video is cached on many servers. On the other hand, the unpopular video "Skiing Hintertux" [13], created and published in Germany, is just cached on the data server in Frankfurt a. M. (DE) and saved on the main server in Mountain View (US). The reason for this is that it is very unlikely that the video is watched. So, caching it in a CDN would use memory on the servers, which is not accessed very much. YouTube, therefore, makes a tradeoff and stores popular video data in the CDN to achieve good performance for as many users as possible and accepts, that for unpopular videos the data-delivery is provided by a server, which is more far away or the main server of YouTube. This makes sense in that way, that only 10% of the most popular YouTube videos draw 79% of views [18]. An explanation for caching "Skiing Hintertux" [13] in Frankfurt a. M. (DE) is, that the video was created in Germany, the access to it mainly comes from Germany and that in Frankfurt a. M. (DE) one of the biggest IXP nodes of the world is, which suggests that

Video	Client City	Client IP	Streaming Server Location	Server IP	Traceroute
Despacito - Luis Fonsi [11] - most popular on YouTube - approx. 6,600 Mio views	Helsinki (FIN)	84.250.184.143	Helsinki (FIN)	62.115.64.97	Helsinki (FIN) → Helsinki (FIN)
	Brisbane (AUS)	45.248.77.172	Perth (AUS)	103.2.118.76	Brisbane (AUS) → Perth(AUS)
	Bochum (DE)	2a05:3e00:c:5::10b3	Frankfurt a. M. (DE)	2a00:1450:4001:9::5	Bochum (DE) → Frankfurt a. M. (DE)
	Chișinău (MD)	178.175.132.28	Bucharest (RO)	195.95.178.176	Chișinău (MD) → Bucharest (RO)
Au Revoir - Mark Forster [12] - popular in Germany - language: German - approx. 50 Mio views	Helsinki (FIN)	84.250.184.143	Helsinki (FIN)	62.115.64.92	Helsinki (FIN) → Helsinki (FIN)
	Brisbane (AUS)	45.248.77.172	Mountain View (US)	74.125.152.8	Brisbane (AUS) → Sydney (AUS) → Mountain View, California (US)
	Bochum (DE)	2a05:3e00:c:5::10b3	Frankfurt a. M. (DE)	2a00:1450:4001:9::5	Bochum (DE) → Frankfurt a. M. (DE)
	Chișinău (MD)	178.175.132.28	Mountain View (US)	74.125.13.148	Chișinău (MD) → Bucharest (RO) → Mountain View, California (US)
Skiing Hintertux [13] - unpopular one - language: Germn - approx. 100 views	Helsinki (FIN)	84.250.184.143	Mountain View (US)	173.194.150.186	Helsinki (FIN) → Mountain View, California (US)
	Brisbane (AUS)	45.248.77.172	Mountain View (US)	74.125.164.137	Brisbane (AUS) → Sydney (AUS) → Mountain View, California (US)
	Bochum (DE)	2a05:3e00:c:5::10b3	Frankfurt a. M. (DE)	2a00:1450:4001:68::6	Bochum (DE) → Frankfurt a. M. (DE)
	Chișinău (MD)	178.175.132.28	Mountain View (US)	173.194.164.170	Chișinău (MD) → Bucharest (RO) → Mountain View, California (US)
Hei rakas - Behm [14] - popular in Finland - language: Finnish - approx 2 Mio. views	Helsinki (FIN)	84.250.184.143	Helsinki (FIN)	62.115.64.94	Helsinki (FIN) → Helsinki (FIN)
	Brisbane (AUS)	45.248.77.172	Mountain View (US)	173.134.28.102	Brisbane (AUS) → Sydney (AUS) → Mountain View, California (US)
	Bochum (DE)	2a05:3e00:c:5::10b3	Frankfurt a. M. (DE)	2a00:1450:4001:32::1	Bochum (DE) → Frankfurt a. M. (DE)
	Chișinău (MD)	178.175.132.28	Bucharest (RO)	195.95.178.176	Chișinău (MD) → Bucharest (RO)
Teekkarihymni [15] - unpopular - language: Finnish - approx 15,000 views	Helsinki (FIN)	84.250.184.143	Helsinki (FIN)	62.115.64.92	Helsinki (FIN) → Helsinki (FIN)
	Brisbane (AUS)	45.248.77.172	Mountain View (US)	74.125.96.7	Brisbane (AUS) → Sydney (AUS) → Mountain View, California (US)
	Bochum (DE)	2a05:3e00:c:5::10b3	Frankfurt a. M. (DE)	2a00:1450:4001:2b::8	Bochum (DE) → Frankfurt a. M. (DE)
	Chișinău (MD)	178.175.132.28	Bucharest (RO)	195.95.178.176	Chișinău (MD) → Bucharest (RO)

Table I. Measurement results for the setup given in II-B.

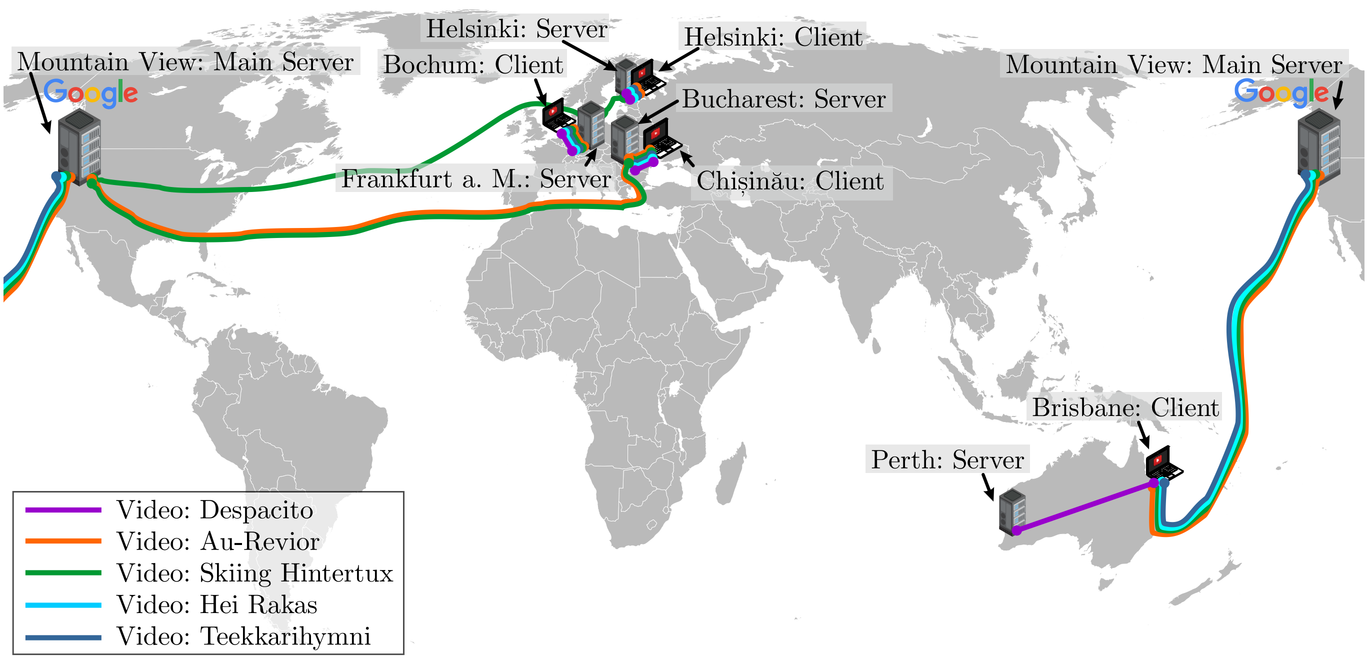

Figure 5. Locations of clients and servers for the measurement results shown in table I.

the CDN is having there big servers [19]. Also, the origin has an impact on the delivery server. As an example, the Finnish music Video "Hei rakas" [14], is probably most viewed from Finland and users within the European Union, but not from Australia. Therefore the video is cached in the CDN within the European Union, but not in Australia and the video is delivered from Mountain View to a client located in Australia. All in all, the CDN is structured in the way that it tries to deliver popular videos to the user with the shortest geographical distance, but also redirects the requests to other servers to achieve load balance in the network and considers the properties origin, language and mostly viewed destinations to not cache a video on a server which is located in a region where only a negligible number of people is accessing the video.

REFERENCES

[1] "Wireshark," accessed February 17, 2020, https://www.wireshark.org/.

[2] "Traceroute," accessed February 19, 2020, https://manpages.debian.org/buster/traceroute/traceroute.1.en.html.

[3] "Alexa internet (amazon web services): Most viewed websites," accessed February 17, 2020, https://www.alexa.com/topsites.

[4] "Alexa internet (amazon web services): Youtube statistics," accessed February 17, 2020, https://www.alexa.com/siteinfo/youtube.com.

[5] "Hotline bling-drake," accessed February 17, 2020, https://www.youtube.com/watch?v=uxpDa-c-4Mc.

[6] "Rfc 8446," accessed February 17, 2020, https://tools.ietf.org/html/rfc8446.

[7] "Decrypting tls browser traffic," accessed February 19, 2020, https://redflagsecurity.net/2019/03/10/decrypting-tls-wireshark/.

[8] S. P. Rao and S. Karvonen, "Experimentation with the youtube content delivery network (cdn)," 11 2013.

[9] H. Nam, K.-H. Kim, and H. Schulzrinne, "Qoe matters more than qos: Why people stop watching cat videos," 04 2016, pp. 1–9.

[10] G. Dimopoulos, P. Barlet-Ros, and J. Sanjuàs-Cuxart, "Analysis of youtube user experience from passive measurements," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, Oct 2013, pp. 260–267.

[11] "Luis fonsi - despacito ft. daddy yankee," accessed February 16, 2020, https://www.youtube.com/watch?v=kJQP7kiw5Fk.

[12] "Mark forster - au revoir (videoclip) ft. sido," accessed February 16, 2020, https://www.youtube.com/watch?v=MtDPKJSsBgc.

[13] "Skiing hintertux," accessed February 16, 2020, https://www.youtube.com/watch?v=VolFDcoFW84.

[14] "Behm - hei rakas (lyriikkavideo)," accessed February 16, 2020, https://www.youtube.com/watch?v=C_FNd9iY9EE.

[15] "Teekkarihymni," accessed February 16, 2020, https://www.youtube.com/watch?v=pWgojCCcwKst=7s.

[16] "Https," accessed February 17, 2020, https://en.wikipedia.org/wiki/HTTPS.

[17] "Submarine cable map," accessed February 16, 2020, https://www.submarinecablemap.com/.

[18] "Pew research center: Youtube statistics," accessed February 17, 2020, https://www.pewresearch.org/fact-tank/2019/12/04/10-facts-about-americans-and-youtube/.

[19] "De-cix," accessed February 17, 2020, https://www.de-cix.net/.