Sarah Tse
ELEC-C7420

YouTube CDN Infrastructure Analysis

Experiment Setup and Goals:

The experiment consists of monitoring the traffic, captured by Wireshark, generated by searching and playing the *Despacito* by Luis Fonsi music video and a vlog that receives less than 20 views per day. Both videos are streamed in 1080p and run for about 4 minutes and 30 seconds. The data is captured at around 5 pm on a Saturday at two different locations: Aalto University in Otaniemi and a home network in Kamppi. Due to travel time, the videos are not played at the exact same moment but within 30-45 minutes of each other.

In theory, higher streamed videos are more accessible around the world because they are stored in servers around the world due to their popularity. In contrast, low streamed videos are not as accessible because the need to stream the video is low. The goal of this experiment is to discover if higher streamed videos are stored in several locations and how the process may differ with a low streamed video. Currently, *Despacito* holds the record for the video with the most views (approximately 6 billion) which makes it the best video for this experiment. In addition, another goal is to discover the mechanisms behind YouTube's search and video streaming functions.

YouTube Architecture:

With Wireshark, I was able to capture the traffic to and from my machine. Based off of the data captured, there was always one address that made up 60-87% of all the source addresses. The unpopular video is expected to have a higher percentage as it is a 10 seconds longer than the popular video. From here, I was able to gather that these are the addresses of the edge servers that delivered the content to my machine.

Location – Video Type	Address	Percentage of total packets captured	IP Location
Home Network – Popular	[redacted]	59.54%	Sweden
Home Network – Unpopular	[redacted]	65.46%	USA
School Network – Popular	[redacted]	81.70%	Sweden
School Network – Unpopular	[redacted]	87.55%	Ireland

Table 1: Most used IP addresses

From here, I worked backwards to determine how the search function works. Fundamentally, on a simpler level, searching is usually done on a database. YouTube utilizes MySQL servers to store usernames and passwords, video thumbnail images, and video file paths. Most popular videos are stored in more data centers around the world to provide faster access to this clients. This is proven by the popular network IP addresses both from the home and school networks. Because IP geolocation lookups are based on the database used, the locations were determined by using several databases and choosing the most frequent result. Another observation that I noticed was that all the traffic utilized the GQUIC transport protocol to stream the video.

Sarah Tse
ELEC-C7420

To determine the architecture of the search function, I looked at the first several hundred IP addresses that had a significant amount of packets and most of them came from the similar addresses.

Home - Popular	Home - Unpop	School – Popular	School – Unpop
Mountain View, CA – Google [redacted] **Finland** [redacted]	**Mountain View, CA – Google** [redacted] **Telia Company AB Finland** [redacted]	**Dublin Ireland - Google** [redacted]	**Dublin Ireland - Google** [redacted]

Table 2: IP addresses from searching for the video

The addresses between the home network and school network are quite different due to the different service providers that is utilized. The home network uses Telia and the school network uses NORDUnet more specifically FUNet in Finland. The red text denotes the same addresses found in the each of the different captures.

```
C:\Users\sarah>tracert youtube.com

Tracing route to youtube.com [2a00:1450:400f:80a::200e]
over a maximum of 30 hops:

  1     1 ms     1 ms    12 ms  [redacted]
  2     1 ms     1 ms     1 ms
  3    67 ms     5 ms     3 ms  espoo1-et-0-1-0-1.ip.funet.fi [2001:708:0:bad:c0de:ffff:ffff:fd]
  4     2 ms     1 ms     1 ms  ndn-gw.funet.fi [2001:948:3:1d::2]
  5    12 ms     8 ms     8 ms  se-fre.nordu.net [2001:948:1:18::2]
  6    11 ms     8 ms     8 ms  se-kst2.nordu.net [2001:948:1:1e::3]
  7    12 ms     9 ms     9 ms  as15169-10g-sk1.sthix.net [2001:7f8:3e:0:a500:1:5169:1]
  8    11 ms    10 ms    13 ms  2001:4860:0:134b::1
  9    10 ms    20 ms     9 ms  2001:4860:0:1::16e9
 10    24 ms    35 ms    21 ms  arn11s03-in-x0e.1e100.net [2a00:1450:400f:80a::200e]
```

Figure 1: Trace Route to youtube.com

Through this analysis, along with further research, I created a map of YouTube's architecture shown in Figure 2. The main connection between all the objects in the diagram is Google Cloud Platform. Because Google is already in the business of providing web applications and internet search services, they have their own infrastructure that acts as a Content Delivery Network(CDN) and as a host to YouTube. YouTube also utilizes Google's authentication system for user signups and authentication.

Sarah Tse
ELEC-C7420

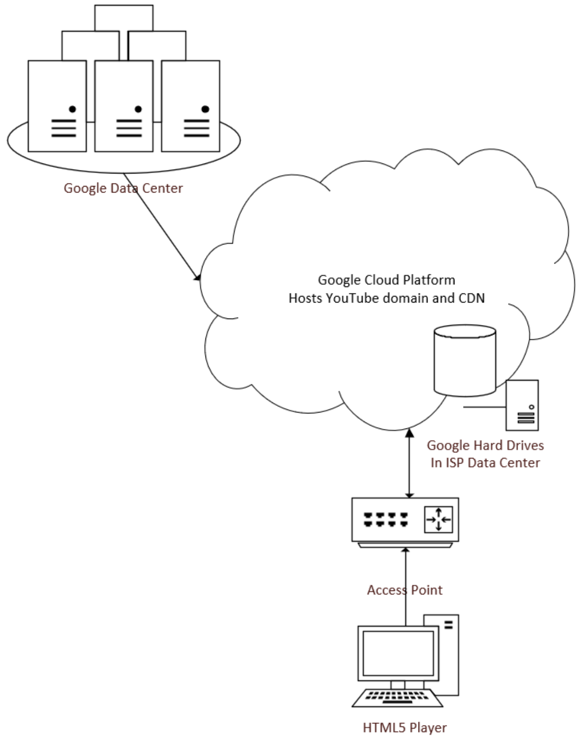

Figure 2: YouTube Architecture

Workflow and YouTube Content Delivery Network

At the beginning of each Wireshark capture, there is a GQUIC (transport layer protocol similar to UDP) 'Hello Client' packet that sets up a connection between the Google server and me, who is the client. When the packet reaches the google server with the application data, a search query is entered into the video link database. Because video files are large and can be inefficient to store in every datacenter, file paths to different servers are used to ensure quick content delivery. In terms of determining which videos to show, YouTube compiles a list of videos utilizing the YouTube algorithm, which is a subprocess that uses metadata and statistics to determine which videos are shown first. Through the GQUIC connection a series of packets were exchanged to deliver the search results webpage. In the Wireshark files, there are no HTTP captures because all website traffic is now secured with HTTPS. Even though I was unable to capture this data, in theory, the application data is encapsulated within the packets sent. The GQUIC packet is then encapsulated with an IP header and so on and so forth.

When the user clicks the video link, another series of exchanges are made to find exactly where the video is stored. This is where YouTube's content delivery network (CDN) starts to work. About 2.5 years ago, a pair of curious YouTubers interviewed several YouTube engineers and created a video that gave a simple framework of how YouTube's CDN worked. First and foremost, hard drives are sent to the biggest data centers of Internet Service Providers around the world. These hard drives are in charge of storing the popular content. If the video is not in these hard drives, a request is made out to Google data centers to find this video. Once the video is found, the video is cached to the client's closest data center and a UDP/GQUIC connection is established to start streaming the video. The video is delivered in small sizes to minimize buffering periods and improve user experience.

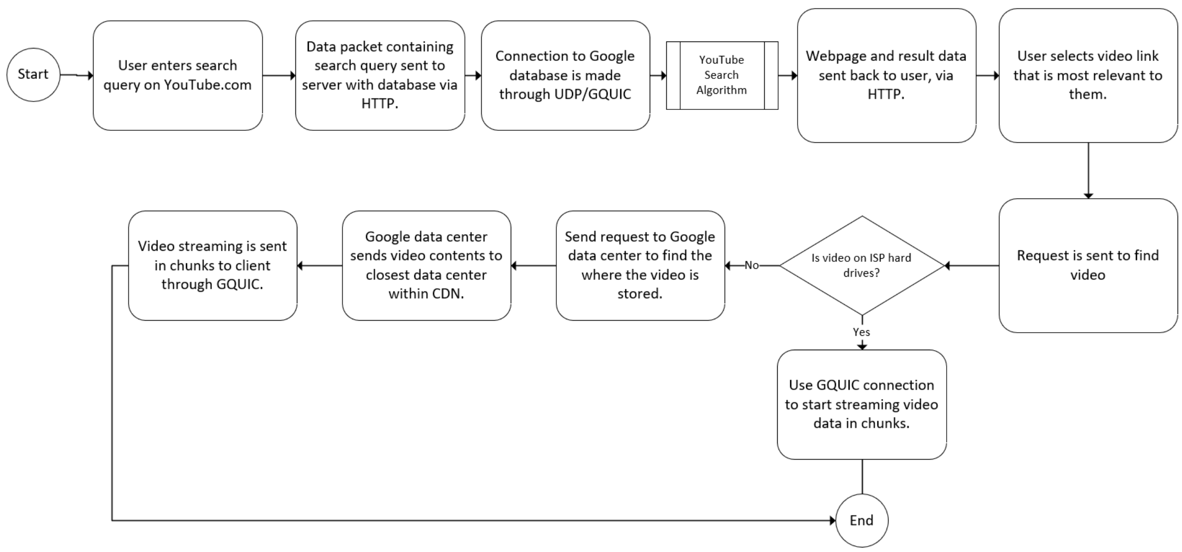

Figure 3: Workflow of search and video delivery functions

References:

Nat and Friends.(2016, Aug 23). *What Actually happens when you watch a YouTube video?* https://www.youtube.com/watch?v=IlFmBbeKRXQ

V. K. Adhikari et al., "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs," in IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 1984-1997, Dec. 2015. doi: 10.1109/TNET.2014.2354262