Exploring YouTube's CDN Heterogeneity

Anh-Tuan Nguyen[1], Olivier Fourmaux[1(✉)], and Christophe Deleuze[2]

[1] Sorbonne Université, CNRS, LIP6, 75005 Paris, France
{anh-tuan.nguyen,olivier.fourmaux}@lip6.fr
[2] Univ. Grenoble Alpes, Grenoble INP
(Institute of Engineering Univ. Grenoble Alpes), LCIS,
26000 Valence, France
christophe.deleuze@lcis.grenoble-inp.fr

Abstract. In this paper, we set up measurements and make performance and geographic analysis of YouTube CDN video platform. We use large distributed testbeds, like PlatnetLab and EdgeNet, to grasp the heterogeneity of client situations. Those facilities can work as real clients without any simulation. From these infrastructures, we generate numerous requests to YouTube video servers. Using a reduced initial set of nodes in different geographic location, we continuously measure information related to YouTube homepage websites and video servers, and calculate the latency from clients to cache servers. We also look at the geographical location of YouTube servers. This enables a better understanding of cache mapping strategy and draws the map of the system. Our first result focus on distance between users and data centers before studying dynamic aspect of the system. The information we collect can be of interest to e.g. ISP network operators who need to improve their network architecture to minimize costs and optimize quality for the user.

Keywords: CDN · Measurement · YouTube · PlanetLab · EdgeNet

1 Introduction

The communication model of World Wide Web was initially designed with the content located and served from a unique server host. Users who want to access these content interact with the client that will generate requests through the Internet to the server. Time to transmit content may be long. Today, the classical structure of the network, especially for the web, has dramatically changed. Internet infrastructure and bandwidth needs have strongly increased, and the types of access have multiplied through the diversity of customers and wireless technologies. To follow this, the structure and content of the web has also changed. Multimedia content and video are growing fast and are used extensively. Websites also use automated settings with scripts from the server, adapting to very heterogeneous users for optimizing content and bandwidth. As a result, the number of users increased so much that websites cannot fit on a unique server

T. Q. Duong et al. (Eds.): Qshine 2018, LNICST 272, pp. 158–166, 2019.
https://doi.org/10.1007/978-3-030-14413-5_13

anymore and led to the development of new load balancing and content distribution systems. Initially, popular contents were simply cloned on many different servers located at many Data Centers in different locations around the world. A user needs to find and pick the closest mirror near him manually and can cause unbalanced situations. Nowadays, content distribution networks (CDNs) such as Akamai [1] or Amazon [2] rely on dynamic mechanisms. They set up dedicated DNS servers inside their networks, automatically analyzing the user's IP address to make adapted answer based on geographic proximity, resource costs, bandwidth availability, and other factors. Content from the origin server is then replicated and stored in or near the Internet provider network (ISP). This infrastructure is not correlated with the routing system and difficult to measure, predict and manage. With the rise of video traffic, we were motivated to run distributed measurement platform to get more information on one of the most popular video content providers: YouTube. Its video delivery is provided by dedicated worldwide CDN and we initiate a measurement campaign to get more understanding about it.

In Sect. 2, we give a brief CDN description and make some emphases on YouTube CDN. In Sects. 3 and 4 we introduce the testbeds used and the measurement we are conducting. In Sect. 5 we provide our first analysis of the Youtube CDN. In Sect. 6, related work is presented. Then we conclude and give some perspective in Sect. 7.

2 Context

To solve the problem of large scale content delivery, *content distribution* architectures have been designed and deployed. A *content distribution network* [6] is made of (among other things) a large number of servers in which the content is made available, plus a *redirection* system whose role is to direct a client to a chosen server. Information about the location of the user, the current load on the servers and on different parts of the network need to be taken into account in order to select the "best" server, *i.e.* the one that will provide the best quality of experience for the user. Also, the redirection needs to be performed *transparently*, without the user being aware of it. Classic approaches are providing different answers to DNS requests depending on the client, encoding client localization in a provided URL. We can also find the use of IP anycast addressing.

Here's a brief description of YouTube content distribution architecture. The user points its browser to a URL such as:

`https://www.youtube.com/watch?v=videoID`

The `www.youtube.com` domain name is resolved to the IP address a *homepage server* and an HTTP request is sent. This homepage server queries a *mapping server* with the client IP address and videoID, that replies with a DNS name (such as `r3---sn-gxo5uxg-jqbe`). This name is used to build a URL that points to the video stream such as:

`https://r3---sn-gxo5uxg-jqbe.googlevideo.com/videoplayback?ei...`

The browser now resolves the domain name to an IP address for a *video server* and gets the video stream from there.

3 Testbed

YouTube CDN is a large-scale distributed system. We consider to building a distributed measurement facility. We need to measure on a large number of clients, then we can have a bigger map of one of the largest CDN system in the world. We decided to choose a stable testbed: PlanetLab; and a very new: EdgeNet. They provides extended access to the system resources: open the socket, send/receive IGMP packets... directly without any system emulation. It allows measurements to be made without any simulator or below another system, which can change the results. With full permission on a machine, we can run unmodified clients, and the result of the measurements are realistic.

PlanetLab [7–9] is a global research network that supports the experimentation of network services. Since the beginning of 2003, more than 1,000 researchers have used PlanetLab to develop new technologies for distributed storage, network mapping, peer-to-peer systems, etc. PlanetLab currently consists of 1353 potential nodes at 717 sites but some of them are no longer maintained. Planetlab is split in several portions and as our Lab, LIP6, run the PlanetLab Europe's control center in Paris and so we chose mostly European sites. Initial measurement started with 27 IP addresses of PlatnetLab Europe nodes spread across Europe in 21 different cities.

EdgeNet [10,12] is a distributed edge cloud, in the family of PlanetLab, GENI, JGN-X, and PlanetLab Europe. It is a modern distributed edge cloud, incorporating advances in Cloud technologies over the past few years. EdgeNet is based on industry-standard Cloud software, with Docker as containerization technology and Kubernetes as the node manager and deployment solution. It is an opt-in global Kubernetes cluster; once a user has authenticated with this portal and been approved, she will be able to use standard Kubernetes tools and technologies to deploy an application across the EdgeNet infrastructure.

4 Measurements

We have developed a "measurement client" written in Python to perform all the steps a real browser would do to display a video. The tool records detailed log of the process including the HTTP request and response messages, the URLs involved at each step, the DNS resolutions performed and the IP addresses found, their geolocation, as well as timestamps for all the important events.

As described in Sect. 2, the host name `www.youtube.com` can be resolved to a number of different *homepage servers*. By repeating requests, we are able to find IP addresses of (some of) these servers. Our tool tries to behave like a normal client, it has a list of video content and manage pause between requests (pause length is chosen randomly in the interval [20, 1200] seconds).

We also knew that the HTML page returned by the homepage server contains URLs for the video themselves, in JavaScript snippets embedded in the page. By parsing the HTML page and the JavaScript snippets, the tool can extract the domain name for the video server (as `r4---sn-gxo5uxg-jqbe.`

googlevideo.com) and resolve it to find the associated IP address. It also queries a geolocation service and records geolocation information for each IP address found (for both video and homepage servers).

Finally, as soon as it has the result of a DNS resolution, the tool measures the round trip time with the server (with the ping utility) and records minimum, maximum and average latency.

5 Results

In the initial phase, we have run clients on 25 nodes of PlatnetLab and 14 nodes of EdgeNet platform, using network of 31 ISPs and covering total 34 cities, in which 12 European countries, 2 countries in North America spanning a total of 10 time zones. We plan to deploy our tool on many more nodes in the future, based on the results obtained from this initial campaign. With 80.000 queries, we collected 356 different IP addresses for the **homepage servers location**. Geolocation information shows that these servers are located in 28 cities of 11 countries. We found that 315 IP addresses are in the United States and Canada and 41 are in Europe. Considering the queries, 73% of the responses came from 20 data centers at different cities in the United States and Canada, and 27% from servers in other countries. Of these, 52% of responses were sent from the client's country. For each request sent to the YouTube website, we observed an average number of 86 HTTP requests and an average amount of 3 megabytes of data transferred. Most of requests finished in less than 6 seconds. In our result, some web servers have an IP belonging to a network supposed in the US but the delay is smaller. There is little mismatch between distance and delay for web servers (especially for PlanetLab nodes), because YouTube has a worldwide network it can allocate its addresses dynamically [15].

Concerning the **video servers location**, we found they are present in 34 cities in all of the 12 considered countries. There are 1173 different IP addresses in our measurement, so each location hosts several video servers. Of these addresses, 1112 are geo-located in the United States and Canada and 61 in Europe. However, contrary to what we saw for homepage servers, most of responses were received from within Europe: 53,8% of responses were sent from an European country, and 46,2% sent from the United States. More importantly, 75,5% of the responses were sent from the client's country. This suggests that geographic distance with the client is (or is strongly correlated with) one of the top criteria YouTube uses to select the video server for a given client.

A response HTML page of YouTube homepage is rather complex. Figure 1 shows the loading of such a response. Each line in this figure is a TCP connection, with purple color in a row is SSL initial connection, gray is stalled, green is time to first byte and blue is actual content loading. Blue vertical line is *DOM-ContentLoaded*, indicating when the initial markup of a page has been parsed and red vertical line is *loaded*, indicating when a page has been fully loaded. This map shows many static components that are mentioned in the following as images, js, css files... After analyzing all URLs from the HTML response of

Fig. 1. Loading timeline map of a YouTube website connection. (Color figure online)

YouTube's server, we classify them into four categories: Homepage, Video cache host names, Static cache host names and unknown type.

The host name `www.youtube.com` is common to all the **homepage servers**. Each DNS resolution for this domain brings the IP address of one of the homepage servers, depending on load balancing of CDN, is serving for location of clients. By using a very strong DNS system inside their network, which is giving us different IP addresses of the same domain, they can redirect us to the most appropriate caching server based on their server selection strategy and load balancing at the connect moment.

A video can be served from many different **video cache servers**. We recorded video cache domains in form:

rα---sn-β-γ.googlevideo.com

In which, α is 1–20, β and γ is alphabetic characters and digits. There is a large number of such video cache domains (we currently found 122), each being mapped to a number of different IPs. This introduces a large amount of flexibility and dynamicity, which can be used to redirect client requests to a video caching server selected to provide the best quality of service at a given time. As an example, we show in Table 1 some of the IPs we found for a precise video hostname (we found 30 different IPs for this name). After receiving our request, the homepage server analyzed and gave us a hostname, client received this hostname, resolved IP and make a connection to the server with this IP.

Inside HTML pages, other domains are visibly dedicated to serving **static content** such as JavaScript code, style sheets and images. Most of these are sub-domains of the `ytimg.com` domain, such as `s.ytimg.com`, `i.ytimg.com`, and `i9.ytimg.com`. The different kinds of static content are classified in the path part of the URL, as sub-directories: `./yts/jsbin/` for JavaScript codes, `./yts/cssbin/` for style sheets and `./yts/img` for images.

Several other types of domain names in response HTML can be observed, but they are not used frequently and their use was not investigated further.

We also study the **latency** and as expected, the average latency time collected for video server requests is smaller than for the homepage server requests. General, the maximum of latency of homepage and video servers is not much different: 196.145 ms for homepage servers and 185.6 ms for video servers. The minimum latency is very low (below 1 ms) for both kinds of servers, suggesting some of our nodes were very close to some YouTube servers. The average latency is much better for video servers (10.47 ms, versus 20.22 ms for homepage servers). This is in line with the result of our geolocation methodology that

Table 1. Some of the IPs & Geolocation of a video cache domain name `r2---sn-gxo5uxg-jqbe.googlevideo.com`

Resolved IP	City	Region	Country
172.217.133.233	Ashburn	Virginia	United States
173.194.164.39	Ashburn	Virginia	United States
173.194.187.9	Ashburn	Virginia	United States
173.194.190.106	Ashburn	Virginia	United States
173.194.190.170	Ashburn	Virginia	United States
74.125.105.122	Mountain View	California	United States
74.125.105.124	Mountain View	California	United States
172.217.130.233	Newark	New Jersey	United States
209.85.230.156	Newark	New Jersey	United States
173.194.5.8	Willard	North Carolina	United States
193.51.224.141	Paris	Île-de-France	United States
172.217.133.10	Ashburn	Virginia	United States

servers in the same country with clients have lower latency. We repeated the latency measurement as well as clients replay the video or select a new video. The collected result included some variation latency of requests from the same client come to the same server at a different time. From the plots, we can see the relation between latency and the distance (Table 2).

Table 2. Percentage of number of requests with latency

Server/Latency less than	1 ms	5 ms	10 ms	20 ms	35 ms	50 ms	100 ms
Homepage	2.27%	31.1%	46.3%	67.15%	80.23%	85.55%	98.77%
Video	14.9%	57.1%	77.2%	89.6%	91.9%	95.3%	99.9%

With the result obtained, most of video request came from Europe will be response from a video cache server inside Europe, almost that response sent from the same country with client. So YouTube server selection strategy is seems effective, it not only gave client the nearest geography server but also the videos are being delivered from a preferred data center. In Fig. 2, we can see the distribution of Video Cache Data Centers spread in Europe and concentrate at the east side of USA. Opposite to Video Cache Data Center location, the location of Homepage Cache Data Centers spread from west to east of USA but more concentrate at the center of Europe. The number of Video Cache Data Center is quiet high than homepage (Fig. 3).

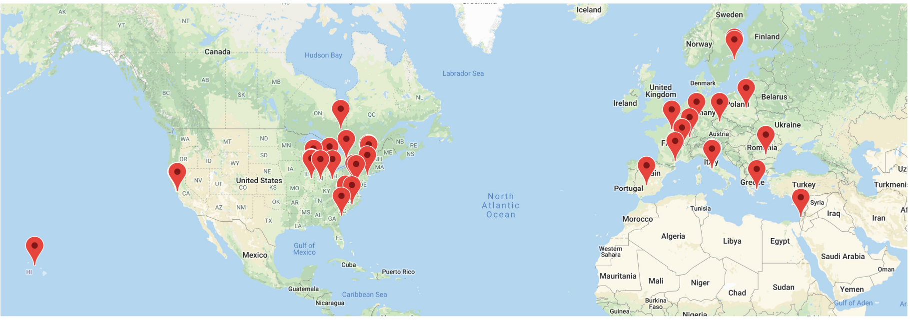

Fig. 2. Distribution of video cache server data center.

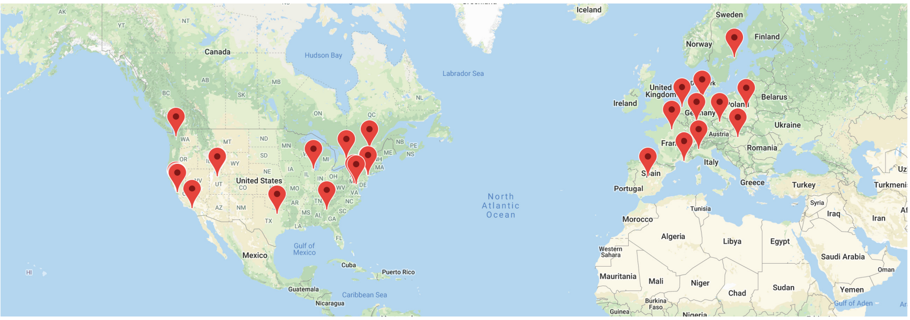

Fig. 3. Distribution of homepage cache server data center.

6 Related Work

There already exists in a number of publications related to YouTube, most of them mainly focus on user behaviors or the system performance. Some other studies make measurement focus on Quality of Service [13]. They found that cache server selection is highly ISP-specific and that geographical proximity is not the primary criterion. In this paper they found YouTube uses HTTP via TCP to deliver the video, and now on UDP with Quic [14]. On the other hand, papers focusing on network-related issues seem less plentiful. A paper focus on DNS, domain name, and latency and throughput measurements are available in [16] for YouTube but a long time ago (2012), so the architecture may have changed. Another paper [17] focus on CDN selection strategy employed by YouTube, and they conclude that at least 10% request will be redirected to a non-preferred server, and the reasons are load balancing, variation DNS servers, load of hot spots due do popular content and unpopular content on given data centers.

7 Conclusion and Perspectives

In this paper we try to get a better understanding of the current CDN spreading structure of YouTube and how video requests are served by their data centers. Our work has been based on datasets collected from the edge of dozens different networks, ISP, located in different countries. Our measurement indicates that the YouTube infrastructure has been changed compared to the one previously analyzed in the publications. In the present system, most of YouTube response are based on geography distance, and they are focusing on video cache servers much more than before. The number of video servers deployed on ISP much more than homepage server. In which, the number of IPs in the United States much higher than in Europe absolutely. Because of this, we need to investigate deeper into their infrastructure to understand how do they do that, and gain their selection strategy. The selection strategy is a very important research, it can give us a big picture of routing and load balancing. We are continuously upgrading our measurement system and expand it geographically, to collect servers at a wider scale. By using the new testbed platform EdgeNet, we will be able to collect more data from the United States and other regions, thereby reinforces the value obtained. This work we have limitations. The main one is the limited reach of clients. All measurements are done by a "client view" which is receiving and analyzing the data sent from YouTube only, it's just an "edge view" of the system. Another point is related to the content that may impact on the server location and that we didn't take into account at this time.

References

1. Akamai. https://www.akamai.com. Accessed 04 Oct 2018
2. Amazon CloudFront. https://aws.amazon.com/cloudfront/. Accessed 04 Oct 2018
3. Azure CDN. https://azure.microsoft.com/en-us/services/cdn/. Accessed 04 Oct 2018
4. Cloudflare CDN. https://www.cloudflare.com/cdn/. Accessed 04 Oct 2018
5. YouTube. https://www.youtube.com/. Accessed 04 Oct 2018
6. Nygren, E., Sitaraman, R.K., Sun, J.: The akamai network: a platform for high-performance internet applications. ACM SIGOPS Operating Syst. Rev. **44**(3), 2–19 (2010)
7. PlanetLab. https://www.planet-lab.org/. Accessed 04 Oct 2018
8. PlanetLab Europe. https://planet-lab.eu/. Access 04 Oct 2018
9. Bavier, A., et al.: Operating system support for planetary-scale network services. In: USENIX Symposium on Networked Systems Design and Implementation, San Francisco (2004)
10. EdgeNet. https://edge-net.org/. Accessed 04 Oct 2018
11. GitHub source code. https://github.com/anh-tuan/YouTube-CDN-measurement. Accessed 14 Nov 2018
12. Cappos, J., Hemmings, M., McGeer, R., Rafetseder, A., Ricart, G.: EdgeNet: a global cloud that spreads by local action. In: Proceedings of ACM/IEEE Symposium on Edge Computing, Bellevue (2018)

13. Hoßfeld, T., Schatz, R., Biersack, E., Plissonneau, L.: Internet video delivery in youtube: from traffic measurements to quality of experience. In: Biersack, E., Callegari, C., Matijasevic, M. (eds.) Data Traffic Monitoring and Analysis. LNCS, vol. 7754, pp. 264–301. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36784-7_11
14. Langley, A., et al.: The QUIC transport protocol: design and internet-scale deployment. In: Proceedings of the SIGCOMM 2017, New York, pp. 183–196 (2017)
15. de Vries, W.B., van Rijswijk-Deij, R., de Boer, P.T., Pras, A.: Passive observations of a large DNS service: 2.5 years in the life of Google. In: Proceedings of Network Traffic Measurement and Analysis Conference (TMA 2018), Vienna, Austria (2018)
16. Adhikari, V. K., Jain, S., Chen, Y., Zhang, Z.L.: Vivisecting YouTube: an active measurement study. In: Proceedings of IEEE INFOCOM, Orlando, pp. 2521–2525 (2012)
17. Torres, R., Finamore, A., Kim, J.R., Mellia, M., Munafo, M.M., Rao, S.: Dissecting video server selection strategies in the YouTube CDN. In: 31st International Conference on Distributed Computing Systems, Minneapolis, pp. 248–257 (2011)