Analyzing the Impact of YouTube Delivery Policies on User Experience

Louis Plissonneau
Orange Labs
France
louis.plissonneau@orange.com

Ernst Biersack
Eurecom
France
erbi@eurecom.fr

Parikshit Juluri
University of Missouri
Kansas City, USA
pjuluri@umkc.edu

***Abstract*—This paper presents an in-depth study of YouTube video service delivery. We have designed a tool that crawls YouTube videos in order to precisely evaluate the quality of experience (QoE) as perceived by the user. We enrich the main QoE metric, *the number of video stalls*, with many network measurements and use multiple DNS servers to understand the main factors that impact QoS and QoE.**

This tool has been used in multiple configurations: first, to understand the main delivery policies of YouTube videos, then to understand the impact of the ISP on these policies and finally, to compare the US and Europe YouTube policies.

Our main results are that: *(i)* geographical proximity does not matter inside Europe or the US, but link cost and ISP-dependent policies do; *(ii)* usual QoS metrics (RTT) have no impact on QoE (video stall); *(iii)* QoE is not impacted nowadays (with good access networks) by access capacity but by peering agreements between ISPs and CDNs, and by server load.

We also indicate a network monitoring metric that can be used by ISPs to roughly evaluate the QoE of HTTP video streaming of a large set of clients at a reduced computational cost.

I. Introduction

Nowadays, web-driven content represents about half of the Internet traffic due to the decrease of P2P and the surge of video sharing sites [13], [10], [11], [12], with YouTube being the most popular. Among the different online video services, HTTP video streaming (using Flash technology) is the most popular one. Services such as blogs and social networks are also enabling users to embed personal videos, thus expanding the video streaming audience. As the users usually watch HTTP video streaming while downloading therefore called *Progressive Download*, the impact of quality degradation is directly perceived. It is therefore of primary interest for ISPs.

In this paper, our goal is to understand the YouTube distribution policy and its impact on Quality of Experience (QoE) from an end-user's point of view. For this purpose, we designed an active measurement tool to evaluate the QoE of YouTube videos with the number of stalls in the video as primary indicator. The purpose of our work is to shed light on YouTube video delivery policies and its infrastructure.

In over a year, we collected several measurements with different configurations. In order to compare results between ISPs, we took advantage of a multi-connected lab with accesses to different ISPs but with similar access rates. We complemented these findings with some specific European measurements. We also benefited from simultaneous measures collected by a class of Kansas City students that allowed us to compare the distribution policies between Europe and US.

The main results are that video delivery policies vary a lot even inside the same country and for the same geographical location. The traffic distribution is highly dependent on the ISP and this can impact the end-user's QoE. Finally, the network and server load-balancing policies dictate the choice of the cache site used by YouTube to deliver the video, whereas the geographical location is not as important. We have also seen that these policies are very volatile and can change abruptly, even on a timescale of a few months.

This paper is organized as follows: we first review related work in Sect. II. We present our tool and methodology in Sect. III. The datasets used are explained in Sect. IV. The main results are exposed in Sect. V, and the knowledge gained on the YouTube infrastructure is explained in Sect. VI. We conclude the paper in Sect. VII.

II. Related Work

We can divide the measurement studies of YouTube into two groups: passive and active studies. Our work is an active measurement whose goal is to infer QoE as seen by the end-user, and also to understand how YouTube video distribution works. We briefly recall the main passive and active studies of YouTube, before showing how our work differentiates from the other active measurement analyses.

A. Passive Measurement Studies

Many articles have focused on determining the characteristics of YouTube videos. Studies more relevant to our work investigated the YouTube CDN architecture and network related performance either based on NetFlow statistics [2] or on packet captures [13], [20], [5]. The influence of traffic management between ISPs and CDNs is underlined in [6].

The impact of DNS on CDN network performance has been studied in [15] showing that a cooperation between ISPs and CDNs could be beneficial to both parties and to the end-users as well. The importance of DNS resolution has also been studied in [20], where the YouTube server selection is explained and how this knowledge can be applied to design complex load-balancing techniques.

The user experience and the impact of network performance on user behavior has been studied in [7], [14], [13] based on packet traces captured at an ISP. The main results are that usually the default video configurations are used and users often jump within the videos. With good network conditions, this may lead to a large amount of wasted bytes (downloaded but not watched). On the user side, the response to deteriorated network performance is to shorten their video watching sessions.

B. Active Measurement Studies

In [17], PlanetLab nodes were used to probe and compare the server infrastructure of 3 different HTTP video streaming services (including YouTube) by comparing the time taken to download the first MByte of the videos. They investigated the service delay distribution according to the geographical location of users and the characteristics of the video (age and popularity).

PlanetLab based active measurements were also used in [3] to understand the dynamics of YouTube video server selection by studying the mechanisms of load-balancing (static, semi-dynamic through DNS and dynamic through HTTP redirect). In [1], the authors pursue the investigation of the YouTube infrastructure, and give many insights into the YouTube video cache server hierarchy.

The impact of DNS resolvers has been compared in terms of latency and caching in [4] (not specific to YouTube). Application level monitoring for ISPs (a goal that we also share) was studied in [19], not only with applications such as quality evaluation but also routing policy management.

The study of YouTube QoE was undertaken with a crowd-sourcing approach in [8]. This paper shows that the primary QoE factors in YouTube video watching are the number of stalls followed by their duration. In [18], an estimator of a YouTube video QoE was designed to be able to predict future stall events.

C. Novelty of our Work

Our work differs from others since we are not only interested in the network performance to access the YouTube video streaming servers but also in the perceived video playback quality. One work towards these goals asked for manual user feedback via crowd-sourcing [8]. In this perspective, one of the main interests of our study is to automatically estimate the QoE of the users without having to ask users for their feedback. We also present a metric to objectively estimate the QoE based on download throughput and encoding rate.

The difference in our dataset and PlanetLab based measurements is also important. We primarily focus on residential accesses that have different characteristics when compared to PlanetLab accesses (often behind large Universities or Research Centers). In [1], they show that the access rates for Planetlab nodes are significantly higher than residential accesses. This leads to different treatment in the YouTube video delivery. The diversity of ISPs in our data allows us to show that the delivery policy (mainly video server selection Sect. V-A) highly depends on the ISP. Finally, many changes observed during the duration of our measurements (almost one year) show that the YouTube infrastructure is highly dynamic even at the limited timescale of a few months.

III. Methodology

The ability to measure QoE of HTTP video streaming is important as it represents a large part of Internet traffic. Passive monitoring can be used to easily monitor a large set of users, but in this case, the perturbations between the probe and the end-user are not taken into account in the analysis. Moreover, as the video data transfered during HTTP video streaming can become huge, large scale passive monitoring would need too much processing. Therefore, we have chosen to monitor the HTTP video streams directly from the end-user's computer.

The interruptions to video playback can be attributed only to insufficient network conditions in HTTP streaming. The video quality depends only on the encoding. Once the definition has been chosen (by default on YouTube: 360×640), no other image degradation is possible (*e.g.* pixeling). Thus, we have chosen only to decode the timestamps of the Flash Video (FLV) frames. This allows us to have a precise evaluation of video time without the cost of video decoding. With this information, we have **reverse engineered the YouTube flash video player** to model its behavior.

A. Tool Presentation

We have designed a tool, `Pytomo` [16], to measure QoS and QoE of YouTube videos. Our tool functions as follows: After the bootstrapping phase, where we collect the URLs of the most popular videos of the week, we process each URL as follows: *(i)* retrieve the URL of the video server; *(ii)* perform the DNS resolution to obtain the IP address of video server; *(iii)* collect QoS statistics ; *(iv)* collect QoE statistics. Videos related to the current video (obtained through YouTube API) are then added to the list of videos to be crawled.

1) Network Statistics: Our tool collects the following statistics (see [9] for a detailed description):

- Ping statistics: min, max, average (over 10 packets)
- Video information: format, duration, length, mean encoding rate
- Download statistics: average throughput, initial throughput (over the first 3 seconds), maximum instantaneous throughput (for a TCP session)

These network statistics are collected per video for each IP address of the video servers.

2) Model of Video Playback: The goal of this model is to be able to detect and count interruptions in the streaming video playback. A large scale QoE study on YouTube quality [8] has shown that **interruptions (stall) in the videos are the main Quality of Experience (QoE) indicator for video streaming**. At the time of writing, seamless video rate adaptation was not available on the main progressive video sites, such as YouTube. Thus, the only way to cope with reduced network throughput was to wait for more data. To model the streaming video playback, we maintain two metrics:

$D(t)$: Amount(seconds) of video content downloaded up to time t *i.e.* the amount of video that is downloaded in terms of playback duration (obtained through the timestamps of FLV tags)

$P(t)$: Amount (seconds) of the video utilized up to time t in terms of playback duration *ie.* the amount of video that was watched.

These two timescales correspond respectively to the gray and red bars in the YouTube player. Obviously when the red bar corresponding to the playback gets close to the gray bar corresponding to the downloaded video, the playback is interrupted. Thus, we have

$$D(t) - P(t) < \text{minimal-playout-buffer} \Rightarrow \text{Playback stops}$$

The restart of the playback occurs when the amount of video that has been buffered is enough.

$$D(t) - P(t) > \text{minimal-restart-buffer} \Rightarrow \text{Playback resumes}$$

By keeping track of the state of the playback, we were able to infer the number of interruptions during video playback. This model does not take into account jumps inside the video or playback pauses initiated by the user. We are aware of these limitations and think this model should reflect common user behavior. Moreover, in case of a jump inside the video, the model is still valid: as shown in [14], a jump in a part of the video that is not already downloaded creates a new connection starting at the requesting time (instead of beginning).

The video playback statistics that we collected were the initial buffering duration, the number of interruptions, the total buffering duration and the seconds buffered at the end of the download.

In Sect. III-B we explain how the values for the model are obtained.

3) Design Implications: As we begin our crawl with the most popular videos of the week (by default), we were biased towards popular videos. This is a deliberate attempt so as to assess the QoS and QoE for the content that most users watch. In [1], the authors show that *cold* (unpopular) videos are much more likely to encounter HTTP Redirect, mainly due to cache miss in the video datacenter. This implies that in our case, HTTP Redirect should be due to a high video server load (and not cache miss).

B. Validation Process

In order to obtain reliable results, the validation and calibration of our tool was carefully undertaken. We use a local server to deliver the video so that we could completely control the video delivery during the calibration process. We simultaneously launched a video download with our crawler and a video playback in a browser. The video was delivered by our local server; note that we had to use a proxy for the video player in the browser since the domain security parameters in the YouTube Shockwave player do not allow queries on domains other than `youtube.com` (such as `localhost`). The total control of the video server allowed us to precisely check the threshold values for the various parameters of the model (see III-A2). We gradually varied the minimum amount of data initially transferred to determine the precise value for the amount of data required to start the video playback. We used a similar approach to measure the other parameters. Therefore we visually verified that the playback modeled in the tool corresponded to the playback in the browser. The following values were determined

- Seconds of video content initially buffered (*initial buffer*): 2.0 seconds;
- Seconds of video content needed to continue playback (*minimal playout buffer*): 0.1 seconds;
- Seconds of video content needed to resume playback (*minimal restart buffer*): 1.0 seconds.

These values agree with the ones chosen in [18] that infer the video quality based on browser events. Decoding the FLV timestamps allowed us to determine precise values of these parameters.

IV. Datasets Details

Our tool is able to run on any PC with minimal setup, thus enabling us to run it under various environments.

1) Volunteer Crawls: We have a large number (145) of volunteer crawls done by ourselves, many colleagues and friends in Europe and the US. This has allowed us to first test our tool, and then to have many different vantage points for analysis. Theses crawls started in March 2011 and are still running at the time of writing (February 2012). Their durations varied from a few hours to many days.

2) Controlled Crawls: We have also benefited from a set-up at a single location connected to different ISPs that provide 7 ADSL, 1 Fiber, and 1 Cable Internet access. Note that the ADSL accesses have exactly the same access bit-rate. These controlled crawls have been useful to launch specific tests across multiple ISPs with comparable setups where only the ISP was varied (geographical location and bit-rate were the same). We focused on 8 crawls with each lasting for at least two days and were obtained between September 2011 to January 2012. We have presented data from only one crawl in September 2011 and one in December 2011 since these are the only two datasets where a significant amount of video stalls occur. This indicates that

- the quality of YouTube videos is highly dynamic;
- at the time of measurement, only a few video stalls were observed overall.

These crawls were used to compare YouTube's policy with respect to the ISP without any difference in the access links.

3) Kansas City Crawls: Finally, a complete class of UMKC students were assigned to run simultaneous crawls from their homes on weekdays in the second week of December 2011 for two hours. These 70 Kansas City crawls gave us some insights into the US market. For YouTube, the US represents about 15% of YouTube traffic and 28% of YouTube users.

These are useful when comparing the findings from the European and US crawls.

Note that the instantaneous throughputs recorded in the crawls allowed us to validate that no access network limitation was encountered; either by the access rate limit, or by excess usage of other applications while using our crawler.

V. Results

In this section, we expose the main results from our experiments. We show the impact of the DNS server used and of the ISP on the selection of the IP address and video server respectively. We show that these two key components have an unexpected impact on the QoS of video streaming.

A. Video Server Selection

1) YouTube Video Server URLs: The URL of a YouTube video is usually: `http://youtu.be/XXXXXXXXXXX` or `http://www.youtube.com/watch?v=XXXXXXXXXXX`. The YouTube video webpage comprises of multiple parts: the main video in the flash player and the rest (comments, related videos, ads…). The video played in the flash player is downloaded using another TCP connection. This connection is responsible for the video delivery but our analysis focuses only on the connection to the video server.

The URL of the video server is customized according to the IP address of the requesting user. We have listed the main types of URLs in Tab. I for controlled crawls (in France) and in Tab. II for Kansas City crawls (in the US). We adopted the same naming convention as in [1]. The 2 main types of URLs in our data are: `lscache` and `nonxt`. They represent primary cache locations of the YouTube infrastructure. A city code, corresponding to the local airport code, is always included in the URL and indicates the preferred location of the YouTube cache site. For some ISPs, a specific URL that includes the name of the ISP along with the city code is given; this should direct users to cache sites dedicated to the ISP.

In [1] the authors showed that the mapping of the video ID to the URL of video servers is *fixed*. This means that if a video is served by a primary cache site as `...v6.lscache2...` with one ISP, it can be directed to another primary cache site but with the same `v6` and `lscache2` in the video server URL.

In our data, the secondary and tertiary cache locations of the YouTube infrastructure were used only in the case of redirections. Their URLs are of the form: `...v[1-24].cache[1-8].c.youtube.com`. Note that there are also unicast hostnames to directly address physical servers: `r[1-24].CITY_CODE.c.youtube.com`. We encounter these URLs only in the case of redirection.

2) In Europe: The most common form of video server URL is `lscache` as shown in Tab. I for the controlled crawls. In these crawls from France, the city codes are `par` and `ams` for Paris and Amsterdam, respectively. From Tab. I, this preferred location clearly depends on the ISP. Here are the main findings from Tab. I:

- ISP B had all its video server URLs on one cache site (`par08s01`) in Paris.
- ISP N had all its video server URLs with Paris cache site as the preferred location but with two different logical names (`par08s01` and `par08s05`).
- ISP O had a dedicated cache site (*`ISP_O`*`-par1`), and the IP addresses of this site belonged to a specific AS (36040).

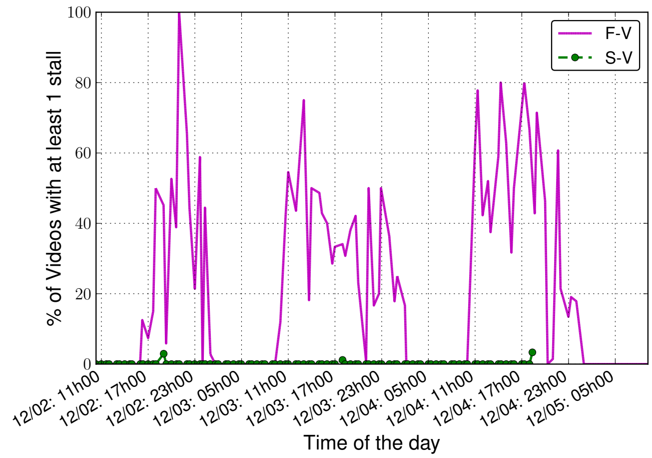


Fig. 2: Evolution of the percentage of videos with at least one stall over time (per each period of 60 minutes) for two ISPs (F-V and S-V) during December 2011 controlled crawl

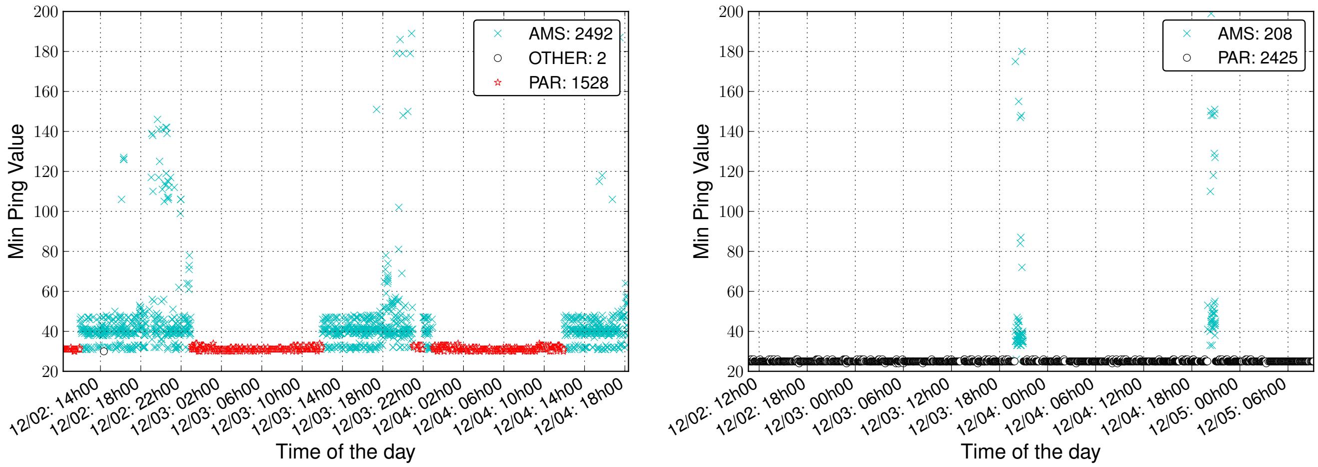
- ISPs S and F were directed to cache sites in Paris or Amsterdam with different proportions: about 66% to Amsterdam for ISP S and 10% for ISP F.

This highlights that the customization of video server URLs is done for each ISP.

The network impact of the location on the ping time was found to be very low. For example, the minimum ping time to the Paris video servers was *23.8* ms and of *28* ms to Amsterdam (because of a relatively small distance between the two cities). But the main point is that the choice of the preferred location is dependent on the time of day as shown in Fig. 1. This indicates a deliberate choice depending on ISP. Moreover, even if the difference on the minimum ping value is very low, the cross traffic on the path from France to Amsterdam can increase the ping value to as high as *200* ms. Fig. 1 shows a large variance in ping times towards the Amsterdam video servers. Overall, the average ping time to the Paris video URLs is of *25.6* ms, whereas it's *53.8* ms to Amsterdam.

3) Impact on QoE: Focusing on the QoE, Fig. 2 shows the average number of interruptions per each period of 60 minutes for 2 ISPs during a controlled crawl in December 2011. This demonstrates that the preferred location had almost no impact on the interruptions. In this crawl, F-V access had lots of periods with many videos affected by stalls, while using video servers based in Paris. Whereas, S-V access had no stalls even though it was mainly served by video servers in Amsterdam. Indeed, the factor that affected the interruptions was the average throughput and not the minor differences in the delay to the server.

At TCP level, a ping time of *200* ms means that 5 TCP windows can be transmitted per second. With a window size of 64 kBytes (minimal value), it leads to a maximum throughput of *320* kBytes/s. The average encoding rate of videos in our data was *555* kb/s or about *70* kBytes/s. This means that this



(a) S-V

(b) F-V

Fig. 1: Ping time in milli-seconds to main YouTube cache sites observed in a controlled crawl in December 2011

URL Regexp	A	B-A	B-F	F-R	ISP F-V	N	O-L	S-E	S-V
o-o.preferred.par08s01.v[1-24].nonxt[378].c.youtube.com	0	1	2	0	0	0	0	0	0
o-o.preferred.ams03g05.v[1-24].nonxt[378].c.youtube.com	0	0	0	0	2	0	0	0	4
o-o.preferred.par08s01.v[1-24].lscache[1-8].c.youtube.com	0	2676	2677	0	0	1890	0	1967	1528
o-o.preferred.par08s05.v[1-24].lscache[1-8].c.youtube.com	1636	0	0	952	2425	799	0	0	0
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	150	0	0	0	206	0	0	3033	2488
o-o.preferred.*ISP_O*-par1.v[1-24].lscache[1-8].c.youtube.com	0	0	0	0	0	0	2591	0	0
other	0	0	0	0	0	0	1	0	2

TABLE I: Number of videos for each ISP according to Regexp on a video server URL for a controlled crawl in December 2011

delay to the video server allows, when there is no congestion (losses), a throughput largely above the one needed to achieve playout without stalls.

In summary, the **video server selection clearly depends on the user's ISP and mainly obeys engineering and load-balancing considerations** rather than the *closest source* or similar strategies.

4) In the US: We used the Kansas City crawls to compare the knowledge gained while analyzing the European data with that of the US. The large number of URLs of video servers (14) indicated that YouTube allowed many more cache sites to be included in the distribution of videos in the US than in Europe. In Tab. II, for each video server URL regexp, we counted the number of videos and the average ping time to the servers. Note that we only included the data from the main prefix (/24) for each URL regexp.

In Tab. II, most of the cache sites are located in the West and Mid-West regions of the US. The most frequent location is Washington DC, even if it is about twice as far (ping-wise) as Dallas. We also had some video servers that were far from Kansas City, such as Miami. If we closely look at the ping values for some cities, we had some variable results. For example, Dallas cache sites had approximately 24 ms or about 50 ms ping values. The reason was that some caches at the Dallas sites had IP addresses in the Google AS (15169) and others in the YouTube AS (43515). The path towards these distinct ASes can thus be different, resulting in different ping times. Note that in our data, the larger ping times correspond to cache sites in the YouTube AS. This was the case for both Dallas and Washington DC.

This validates that the proximity of the cache site plays only a secondary role in video server selection, and that interconnection between ISPs and ASes is a primary factor in network performance.

B. Evaluation of QoE Approximation Techniques

We tested how *precise* could be an approximation of video stalled by an indicator relating download throughput and encoding rate. This could be useful for large scale analysis where complete analysis of the download is not possible. We explored two metrics: one based on Deep Packet Inspection (DPI) and the other based only on flow level statistics.

1) DPI Metric: We have chosen the same metric as in [13]:

$$\text{reception\_ratio} = \frac{DownloadThroughput}{VideoEncodingRate}$$

URL Regexp	Location¶	Nb. of samples	Average Ping time
	WEST		
http://o-o.preferred.iad09g05.v[1-24].lscache[1-8].c.youtube.com	Washington DC	1439	97
http://o-o.preferred.sjc07s11.v[1-24].lscache[1-8].c.youtube.com	San Jose	446	73
http://o-o.preferred.lax04s12.v[1-24].lscache[1-8].c.youtube.com	Los Angeles	147	75
http://o-o.preferred.iad09s12.v[1-24].lscache[1-8].c.youtube.com	Washington DC	44	60
http://o-o.preferred.sjc07s15.v[1-24].lscache[1-8].c.youtube.com	San Jose	10	61
	MID-WEST		
http://o-o.preferred.comcast-dfw1.v[1-24].lscache[1-8].c.youtube.com	Dallas	719	50
http://o-o.preferred.dfw06g01.v[1-24].lscache[1-8].c.youtube.com	Dallas	308	59
http://o-o.preferred.dfw06s08.v[1-24].lscache[1-8].c.youtube.com	Dallas	190	24
http://o-o.preferred.mna-mci1.v[1-24].lscache[1-8].c.youtube.com	Kansas City	71	184
http://o-o.preferred.ord12s01.v[1-24].lscache[1-8].c.youtube.com	Chicago	64	1105
http://o-o.preferred.kanren-lwc1.v[1-24].lscache[1-8].c.youtube.com	Lawrence	50	38
	EAST		
http://o-o.preferred.mia05s05.v[1-24].lscache[1-8].c.youtube.com	Miami	660	261
http://o-o.preferred.lga15s20.v[1-24].lscache[1-8].c.youtube.com	New York	89	53

TABLE II: Ping times according to video server URLs for Kansas City crawls

¶ we mention the city corresponding to the airport code inside the URL

A download throughput lower than an encoding rate should result in interrupted playback (reception ratio < 1). In this case, one can use a DPI tool to retrieve the video encoding rate from the video streaming flow.

To evaluate the accuracy of this method, we used two standard metrics usually used in classification studies(as in [11]). They are based on the concepts of

- *True Positive* ***TP****:* reception ratio > 1 and the video had no stall
- *False Positive* ***FP****:* reception ratio > 1 but the video had at least one stall
- *True Negative* ***TN****:* reception ratio < 1 and the video had at least one stall
- *False Negative* ***FN****:* reception ratio < 1 but the video had no stall.

Out of these notions, we built the following evaluation metrics:

- **recall** = TP / (TP + FN): This corresponds to the fraction of uninterrupted videos correctly evaluated.
- **precision** = TP / (TP + FP): This corresponds to the ratio of uninterrupted videos in the videos with reception ratio > 1.

Here are the results of the F-V December 2011 crawl: 91.8% of recall and 88.5% of precision. This means that the reception_ratio, based on the video encoding rate, is quite accurate to determine stalls in the videos.

2) Pure-Network Metric: If we further explore the idea of computationally efficient evaluation, we can construct another metric without any DPI phase. We compared the download throughput to the default encoding rate. We measured this default encoding rate at $555kb/s$ in our data. Hence, the metric is:

$$\text{simple\_reception\_ratio} = \frac{DownloadThroughput}{555\,\text{kb/s}}.$$

This leads to the following evaluation of the metric: 28.7% of recall and 100% of precision. This means that this non-DPI metric can surely assess that a video is interrupted, but would class a lot of interrupted videos as good ones.

URL Regexp	# /24
o-o.preferred.par08s0[15].v[1-24].lscache[1-8].c.youtube.com	1
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	4
o-o.preferred.*ISP_O*-par1.v[1-24].lscache[1-8].c.youtube.com	2

(a) September 2011

URL Regexp	# /24
o-o.preferred.par08s0[15].v[1-24].lscache[1-8].c.youtube.com	1
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	12
o-o.preferred.*ISP_O*-par1.v[1-24].lscache[1-8].c.youtube.com	2

(b) December 2011

TABLE III: Distribution of IP prefixes (/24) of video servers of all ISPs for controlled crawls

3) Application of these Metrics: The conclusion of this evaluation was that to **roughly evaluate video streaming QoE**, we can focus on network throughput (instead of parsing all the FLV timestamps) but **a DPI engine was needed** to have a precise evaluation of the encoding rate of the video. Another advantage of this method is that we do not need to be at the end-user side (as the throughput is limited from end to end by TCP). Hence it can be applicable to monitoring probes that are placed in the core network (thus connecting a lot of clients).

VI. YouTube Infrastructure

From the knowledge gained in Sect. V, we tried to gain more insight into the YouTube infrastructure. In this section, we use the controlled crawls but do not separate data per the ISPs as we are interested in the global YouTube infrastructure.

A. Datacenter sizes

1) URLs to IP prefix mapping: The YouTube video servers with the same /24 IP prefix usually sharing the same location. In Tab. III, based on data from our controlled lab, we indicated that the main URL regexps were the number of prefixes /24

URL Regexp	# IPs
o-o.preferred.par08s01.v[1-24].lscache[1-8].c.youtube.com	160†
o-o.preferred.par08s05.v[1-24].lscache[1-8].c.youtube.com	160†
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	328
o-o.preferred.*ISP_O*-par1.v[1-24].lscache[1-8].c.youtube.com	98

† these 160 IP addresses are the same

(a) September 2011

URL Regexp	# IPs
o-o.preferred.par08s01.v[1-24].lscache[1-8].c.youtube.com	80‡
o-o.preferred.par08s05.v[1-24].lscache[1-8].c.youtube.com	80‡
o-o.preferred.ams03g05.v[1-24].lscache[1-8].c.youtube.com	494
o-o.preferred.*ISP_O*-par1.v[1-24].lscache[1-8].c.youtube.com	130

‡ two distinct subsets of 80 IP addresses

(b) December 2011

TABLE IV: YouTube Datacenters sizes according to the Number of IP addresses seen for crawls of all ISPs on each URL Regexp

found with the default DNS server of each ISP. First, note that we joined two regexps (`lscache` URL with `par08s01` and `par08s02`) because they share the same prefix. Also, the Paris site had fewer prefixes than the Amsterdam site. Moreover, the number of prefixes used in Amsterdam had grown rapidly in 3 months, from 4 prefixes to 12. An interesting point is that the /24 prefixes were quite dispersed and could not be merged in larger prefixes. Also, the prefixes were distinct between the URL regexps.

Finally, we have to mention that when the QoS (here ping times) were so small, these differences did not translate into QoE differences. And as seen in Sect. V-A2 for F-V, closer videos servers did not guarantee a better QoE.

2) IP address count: In Tab. IV, we counted the number of IP addresses for each video server URL Regexp. For each URL Regexp, we had exactly 192 different hostnames[1] (also seen in [1]). This means that for the Paris datacenter, we had fewer IP addresses (160) than hostnames. Also note the volatility in the distribution of IP addresses in September 2011 (Tab. IVa). The 160 IP addresses were **shared** between the two main Paris `lscache` URL regexps, whereas in December 2011 (Tab. IVb), 80 **distinct** IP addresses were assigned to each `lscache` URL regexp.

We also sent ping probes to the missing IPs that belonged to the prefix; there was usually no reply to the TCP ping on these IP addresses. This means the **load-balancing used by YouTube** allows us to **cover most of the alive machines and all of the hostnames** of the datacenter even with a 2 days probing period.

As for the distribution of URL regexps, the video server URL clearly depended on the user's ISP. So in December 2011, for some ISPs, distinct subsets of the video servers prefix were used.

Sect. VI-A1 has shown that the Amsterdam site was larger than Paris in terms of prefixes. This was also the case for the number of IP addresses. For the same amount of hostnames, we had many more IP addresses in Amsterdam than in Paris. We had a 50% increase in the number of IP addresses in Amsterdam from September to December 2011.

[1]this corresponds to the whole range of possibilities

ISP	Percentage of Redirection
A	29.22
B-A	29.58
B-F	30.83
F-R	26.57
F-V	25.33
N	30.19
O-L	12.69
S-E	49.02
S-V	45.99

TABLE V: Percentage of Redirection per ISP for December 2011 controlled crawl

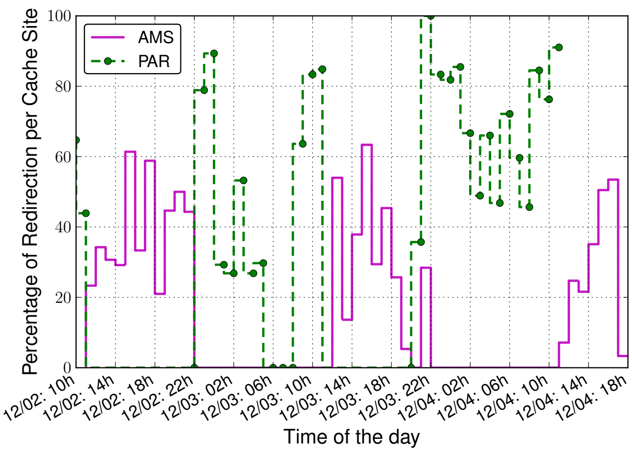


Fig. 3: Percentage of Redirection (over all videos) per YouTube Cache Site for ISP S-V per hour

B. Redirections

YouTube uses redirections to add a level of dynamic load-balancing (on top of the DNS policy and cache site selection according to ISP, which are both centralized). The number of redirections in our controlled crawls were quite variable. In Tab. V, some ISPs had up to 50% of videos redirected while others had only 12% of them. Moreover, these **redirections occured throughout the day** and not specifically at peak hours. The redirect URLs mainly have a unicast form (see Sect. V-A1). In our data, the redirections were usually sent to these cache sites: `par` (Paris), followed by `ams` (Amsterdam), `fra` (Frankfurt), and `lhr` (London).

In [1], [20], the explanation for redirections lies in the unavailability of the requested video or in the datacenter load. Due to our choice of bootstrapping the crawls on popular videos, the chances to have had redirections, because of cache misses were are unlikely. As for the load, the redirections also occur during off-peak hours.

From Tab. V we see that the probability of redirection is dependent on the ISP. In the case of an ISP with a customized URL (like ISP O), there were only 12% of redirections, whereas in the case of ISP S, half of the videos encountered redirections. This is surprising as the cache sites was shared between ISPs. In Fig. 3, we ploted the distribution of redirection over time for the same ISP shown in Fig. 1a. The graph does not show any correlation between time of day (peak *vs.* off-peak hours) and the percentage of redirections. Also, the redirections were usually sent to another cache site; in this case, mainly to London (34%), Frankfurt (30%) or Paris (26%). This means that even if the distribution policy sends a user to an Amsterdam cache site, the redirections could have sent him back to Paris. We conclude that the primary focus of HTTP redirection (except for cache misses) is to unburden the YouTube infrastructure.

So this seems to indicate that **the centralized distribution policies (through cache site selection and DNS) addresses the traffic load balancing, whereas the decentralized distribution policies (through HTTP redirects) addresses the server load.**

VII. Conclusion

We have presented a reliable tool to automatically evaluate the playback quality[2] of YouTube videos as experienced by users. One of the main objectives of this tool is to understand the delivery policy of YouTube and relate it to the DNS resolution policy.

In our study, we used many volunteer crawls to infer the main delivery policies of YouTube videos. We have completed these crawls with controlled crawls in a specific lab to show the difference in treatment between ISPs for accessing the same service. Finally, we used many simultaneous crawls from Kansas City in the US to comment on the difference in infrastructure between Europe and the US.

The main findings of our study are that geographical proximity does not really matter inside Europe or the US, but network/server load-balancing and ISP-dependent policies do. Usual QoS metrics (RTT) have no impact on QoE (video stalls). The number of HTTP redirects are quite high in our data, indicating a globally high load on the YouTube video servers. Finally, QoE is no longer impacted by access capacity but by peering agreement of ISPs and by the server load.

The general conclusion is that YouTube, and more generally the CDNs have many ways to control the content delivery.

1) By customizing the URL of the video server, which is done by the YouTube front-end servers (Sect. V-A)
2) By resolving the URL of the video server to a different IP address, which is done by the YouTube authoritative DNS server
3) By using HTTP redirect messages at the video server level, which is done at the cache site level (Sect. VI-B).

Note that the HTTP redirect messages usually occur when the server decides not to serve the request (*e.g.* when the server is too loaded). Thus, this is a decentralized process. On the contrary, the URL customization and the specific DNS resolution can be controlled centrally. Therefore, we would like to emphasize that YouTube has a large number of knobs to decide what server and what AS a particular video gets served from. From our data, it seems that the primary goals of the video delivery is to use *best* paths and to spare infrastructure.

Moreover, as the routing modifications are usually not advertised by YouTube to ISPs, this may lead to sub-optimal infrastructure usage. A collaboration between YouTube (and more generally the CDNs) and ISPs is therefore needed to use the Internet at its full potential and for the benefit of end-users.

From an operational point of view, we have shown that a network metric (download throughput) and a minimal DPI engine (to retrieve the video encoding rate) can lead to satisfactory results in evaluating the video QoE of HTTP video streaming. This can be efficiently used to monitor the perceived quality of a large number of clients from a central point.

[2] which is much more complex than raw throughput measure

References

[1] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang. Vivisecting YouTube: An Active Measurement Study. In *IEEE INFOCOM 2012 Mini-conference*, 2012.

[2] V. K. Adhikari, S. Jain, and Z.-L. Zhang. YouTube Traffic Dynamics and Its Interplay with a Tier-1 ISP: An ISP Perspective. In *IMC*, 2010.

[3] V. K. Adhikari, S. Jain, and Z. L. Zhang. Where Do You "Tube"? Uncovering YouTube Server Selection Strategy. In *IEEE ICCCN*, 2011.

[4] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing DNS resolvers in the wild. In *IMC*, 2010.

[5] S. Alcock and R. Nelson. Application Flow Control in YouTube Video Streams. *ACM SIGCOMM CCR*, 41(2), April 2011.

[6] A. Finamore, M. Mellia, M. Munafo', and V. Gehlen. Uncovering the Big Players of the Web. In *TMA Workshop*, 2012.

[7] A. Finamore, M. Mellia, M. Munafo, R. Torres, and S. Rao. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *IMC*, 2011.

[8] T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia. Quantification of YouTube QoE via Crowdsourcing. In *MQoE*, 2011.

[9] P. Juluri, L. Plissonneau, and D. Medhi. Pytomo: A Tool for Analyzing Playback Quality of YouTube Videos. In *ITC 23*, 2011.

[10] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *IMC*, 2009.

[11] M. Pietrzyk, J.-L. Costeux, G. Urvoy-Keller, and T. En-Najjary. Challenging Statistical Classification for Operational Usage: the ADSL Case. In *IMC*, 2009.

[12] M. Pietrzyk, L. Plissonneau, G. Urvoy-Keller, and T. En-Najjary. On Profiling Residential Customers. In *TMA Workshop*, 2011.

[13] L. Plissonneau and E. Biersack. A Longitudinal View of HTTP Video Streaming Performance. In *MMSys*, 2012.

[14] L. Plissonneau, T. En-Najjary, and G. Urvoy-Keller. Revisiting Web Traffic from a DSL Provider Perspective: the Case of YouTube. In *ITC Specialist Seminar on Network Usage and Traffic*, 2008.

[15] I. Poese, B. Frank, B. Ager, G. Smaragdakis, and A. Feldmann. Improving Content Delivery using Provider-aided Distance Information. In *IMC*, 2010.

[16] Pytomo. http://code.google.com/p/pytomo/.

[17] M. Saxena, U. Sharan, and S. Fahmy. Analyzing Video Services in Web 2.0: a Global Perspective. In *NOSSDAV*, 2008.

[18] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle. YoMo: A YouTube Application Comfort Monitoring Tool. In *QoEMCS EuroITV*, 2010.

[19] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescape. Broadband Internet Performance: A View From the Gateway. In *ACM SIGCOMM*, 2011.

[20] R. Torres, A. Finamore, J. Kim, M. Mellia, M. Munafo, and S. Rao. Dissecting Video Server Selection Strategies in the YouTube CDN. In *IEEE ICDCS*, 2011.