On the Analysis of QoE-based Performance Degradation in YouTube Traffic

Pedro Casas, Alessandro D'Alconzo, Pierdomenico Fiadino, Arian Bär
FTW - Telecommunications Research Center Vienna
{surname}@ftw.at

Alessandro Finamore
Politecnico di Torino
finamore@tlc.polito.it

Abstract*—YouTube is the most popular service in today's Internet. Google relies on its massive Content Delivery Network (CDN) to push YouTube videos as close as possible to the end-users to improve their Quality of Experience (QoE), using dynamic server selection strategies. Such traffic delivery policies can have a relevant impact on the traffic routed through the Internet Service Providers (ISPs) providing the access, but most importantly, they can have negative effects on the end-user QoE. In this paper we shed light on the problem of diagnosing QoE-based performance degradation events in YouTube's traffic. Through the analysis of one month of YouTube flow traces collected at the network of a large European ISP, we particularly identify and drill down a Google's CDN server selection policy negatively impacting the watching experience of YouTube users during several days at peak-load times. The analysis combines both the user-side perspective and the CDN perspective of the end-to-end YouTube delivery service to diagnose the problem. The main contributions of the paper are threefold: firstly, we provide a large-scale characterization of the YouTube service in terms of traffic characteristics and provisioning behavior of the Google CDN servers. Secondly, we introduce simple yet effective QoE-based KPIs to monitor YouTube videos from the end-user perspective. Finally and most important, we analyze and provide evidence of the occurrence of QoE-based YouTube anomalies induced by CDN server selection policies, which are somehow normally hidden from the common knowledge of the end-user. This is a main issue for ISPs, who see their reputation degrade when such events occur, even if Google is the culprit.

Keywords*—*YouTube; Content Delivery Networks; Performance Degradation; Quality of Experience; Empirical Entropy; Clustering.

I. Introduction

YouTube is the most popular video streaming service in today's Internet, and is responsible for more than 30% of the overall Internet traffic [1], [2]. Every minute, 100 hours of video content are uploaded, and more than one billion users visit YouTube each month[1]. This enormous popularity poses complex challenges to network operators, who need to design their systems properly to cope with the high volume of traffic and the large number of users. The provisioning of YouTube through the massive Google Content Delivery Network (CDN) [9] makes the overall picture even more complicated for ISPs, as the video requests are served from different servers at different times.

CDNs are a vital part of current Internet, as they host a large share of today's Internet traffic [1], [2]. Massively distributed server infrastructures are deployed to replicate content and make it accessible from different Internet locations. For example, Google operates tens of data-centers and server clusters worldwide [9], and deploys thousands of servers inside ISPs, through their Google Global Cache approach[2].

The intrinsic distributed nature of CDNs allows to better cope with the ever-increasing users' content demand. Popular applications such as YouTube are pushed as close as possible to end-users to reduce latency and improve their Quality of Experience (QoE). Load balancing policies are commonly used to limit server load, handle internal outages, help during service migration, etc. Unfortunately, all these control policies are typically very dynamic and the details of their internal mechanisms are not publicly available. The highly distributed server deployment and adaptive behavior of Google's CDN allow for achieving high availability and performance; however, these pose important challenges to the ISPs. The traffic served by CDNs can shift from one cache location to another in just minutes, causing large fluctuations on the traffic volume carried through different ISP network paths. As a result, the traffic engineering policies deployed by ISPs might be overruled by the CDN caching selection policies, potentially resulting in sub optimal end-users' QoE.

Google has recently acknowledged the need of monitoring the content delivery network performance by launching the Video Quality Report (VQR) initiative[3]. Through this service, users can compare statistics related to the perceived quality when accessing YouTube from different ISPs. Interestingly, the only root cause highlighted by such reports is related to limited ISPs bandwidth provisioning. While it is clear that the video service quality is correlated to the available bandwidth, ISPs are not always the only responsible in case of problems. In particular, in this paper we report a case study occurred at the network of a major European ISP, in which sub-optimal server selection strategies adopted by the Google CDN resulted in sharp users' experience degradation[4]. This event shows that actually Google itself might be responsible for YouTube service degradation.

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627, "mPlane". The research leading to these results has been partially performed within the framework of the projects Darwin 4 and N-0 at the Telecommunications Research Center Vienna (FTW), and has been partially funded by the Austrian Government and the City of Vienna through the program COMET.

[1] http://www.youtube.com/yt/press/statistics.html

[2] https://peering.google.com/about/ggc.html

[3] http://www.google.com/get/videoqualityreport/

[4] Conversations with the ISP confirmed that the effect was indeed negatively perceived by the customers.

The most notable recent work related to the understanding of performance degradation events in video distribution from the end user side is [14], where authors conduct a taxonomy of video quality problems using a large-scale dataset of client-side measurements. In this paper we consider the problem of diagnosing QoE-based performance degradation events in YouTube's traffic, using exclusively ISP-based measurements. Through the analysis of one month of YouTube flow traces collected at the network of a large European ISP, we drill down a Google's CDN server selection policy negatively impacting the watching experience of YouTube users during several days at peak-load times. In our study we develop a very simple QoE-based KPI to monitor YouTube videos from the end-user perspective, and use it to identify the aforementioned event. We expect that by explicitly showing that events in which Google server selection policies result in poor end-user experience actually occur, third-party based monitoring initiatives such as the Google VQR would start additionally reporting their own performance.

The insights of our analysis are particularly useful for the ISP, who usually has a hard time in figuring out where are the problems of the service delivery when their customers experience poor performance with YouTube. In the EU project mPlane[5] we are developing a global Internet-scale measurement platform to better understand and diagnose performance degradation events in large-scale services such as YouTube, and this study provides rich input to better develop the measurement and analysis processes.

This paper focuses exclusively on the diagnosis of the aforementioned performance degradation event, and not on its mitigation. The counteractions the ISP and/or the CDN might take upon detection of such events are out of the scope of our study.

The remainder of this paper is organized as follows: Section II provides a brief overview on the papers characterizing YouTube, and those focusing on analyzing performance degradation issues. In Section III we describe the dataset used in the study, and present additional details on the data analysis approach we use, consisting of time-series analysis, entropy-based analysis, statistical distribution-based analysis, and clustering. Section IV presents a characterization of the end-to-end YouTube service as observed from the collected traces, and introduces the QoE-based KPIs for YouTube monitoring. The analysis and diagnosis of the performance degradation in YouTube is performed in Section V. Finally, Section VI concludes this paper.

II. Related Work

The study of the Internet traffic and applications delivered by CDNs has gained important momentum in the last few years [1], [2]. In particular, several studies characterize CDNs architectures and focus on the optimization of their performance, servers location, throughput and latencies [9]–[11]. When it comes to YouTube, its overwhelming popularity and traffic volume have motivated a large research effort on understanding how the service works and performs [3]–[5], covering aspects such as content delivery mechanisms, video popularity, caching strategies, and CDN server selection policies among others.

Some very recent papers tackle the problem of CDN monitoring and detection of performance degradation events in the provisioned services [14]–[16]. In our recent work [15] we have started to study the problem of detecting network traffic anomalies in Internet-scale services provided by major CDNs such as Akamai and Google CDN. In [16], authors present a framework to diagnose large latency changes in CDNs' delivered traffic, and find out that nearly 1% of the daily latency changes observed between users and Google CDN servers increase delay by more than 100 msec. From those latency changes, more than 40% correspond to interdomain routing changes, and more than one-third involve a shift in traffic to different CDN servers. Finally, authors in [14] present a taxonomy of video quality problems using a large-scale dataset of client-side measurements. Among their findings is the observation that about 50% of the observed performance degradation events persist for at least 2 hours, and that between 30-60% are related to the content provider, the CDN, or the client ISP.

III. Dataset and Analysis Approach

The dataset used for the analysis corresponds to one month of YouTube flows, collected at a link of a European fixed-line ISP aggregating 20,000 residential customers who access the Internet through ADSL connections. The complete data spans more than 10M YouTube video flows, served from more than 3,600 Google servers. To identify and diagnose performance issues, we rely on the analysis of the empirical probability distributions of several features describing the YouTube traffic delivery and its performance, such as download throughput, traffic volume served per each observed Google server, etc.. To process the information carried on the probability distributions, we employ entropy as a summarization tool of the empirical PDFs, as well as their inter-distance through an extension of the well know Kullback-Leibler (KL) divergence [12]. In all cases, the study is based on the analysis of the resulting time-series, when considering the temporal evolution of the different features. Finally, we additionally employ unsupervised analysis techniques based on clustering to provide first steps in the unsupervised characterization of the detected problems.

A. YouTube Dataset

Flows were collected from April the 15th till May the 15th 2013. Flows are captured using the Tstat passive monitoring system [18]. Tstat is an Open Source packet analyzer capable of monitoring links up to several Gb/s speed using commodity hardware. Using Tstat filtering and classification modules, we only keep those flows carrying YouTube videos. The complete dataset is imported and analyzed through the DBStream large-scale data analysis system [19]. Finally, using the server IP addresses of the flows, the complete dataset is complemented with the name of the ASes hosting the content, extracted from the MaxMind GeoCity databases[6].

B. Entropy-based Analysis

The sample entropy has been proposed for traffic analysis in multiple contexts, we particularly follow the approach presented in [17]. In a nutshell, given an empirical distribution

[5] http://www.ict-mplane.eu/

[6] MaxMind GeoIP Databases, http://www.maxmind.com.

of a certain variable, its sample entropy captures in a single value a measure of its "shape". More precisely, the entropy of a random variable X is $H(X) = -\sum_{i=1}^{n} p(x_i)log(p(x_i))$, where $x_1, \ldots, x_n$ is the range of values for X, and $p(x_i)$ is the probability that X takes the value x_i. The values of $p(x_i)$ are computed from the empirical probability distributions. Similar to [17], we normalize the sample entropy (between 0 and 1) to the factor $log(n_0)$, where n_0 is the number of distinct x_i values present in a given measurement slot.

C. Temporal-similarity Analysis

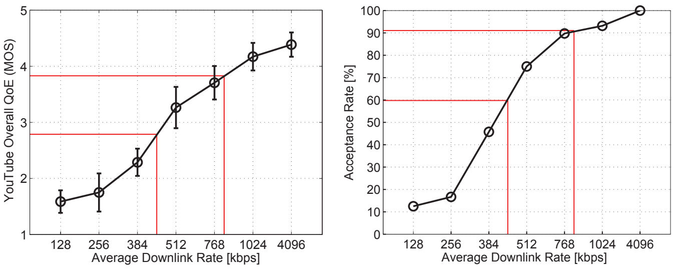
Another approach to summarize changes in the distribution of a certain variable is by computing the KL divergence. Given two probability distributions p and q defined over a common discrete probability space, the KL divergence provides a non-negative measure of the statistical similarity between p and q. To visualize and quantify the degree of (dis)similarity of a large number of distributions over days and even weeks, we use an ad-hoc graphical tool proposed in [12], referred to as Temporal Similarity Plot (TSP). The TSP allows pointing out the presence of temporal patterns and (ir)regularities in distribution time series, by simple graphical inspection. The TSP is a symmetrical checker-board heat-map like plot, where each point $\{i, j\}$ represents the degree of similarity between the distributions at time bins t_i and t_j. In the following analysis, we use the TSP to better depict changes in the sever selection policies used by Google to serve YouTube videos.

D. Unsupervised Analysis through Clustering

The final analysis technique we employ in the analysis is clustering. The objective of clustering is to partition a set of unlabeled patterns into homogeneous groups of similar characteristics, based on some measure of similarity. Our goal is to verify how feasible it is to identify the occurrence of the analyzed performance degradation event in an unsupervised manner. In particular, we aggregate traffic per server IP on a temporal basis, and define a set of traffic descriptors characterizing the behavior of each server. By using the well known DBSCAN clustering approach [20], we show that it is possible to identify the presence of the QoE-based degradation event in the set of server IPs providing the videos. DBSCAN is a powerful density-based clustering algorithm that discovers clusters of arbitrary shapes and sizes, and it perfectly fits our unsupervised traffic analysis, because it is not necessary to specify a-priori difficult to set parameters such as the number of clusters to identify. We use a simple auto-calibration approach to define the required inputs used by DBSCAN, similar to [21].

IV. Quality of Experience and Traffic Characterization

Even if the download throughput has a direct impact on the performance of YouTube provisioning [6], our previous studies [7], [8] have shown that the main impairment affecting the QoE of the end-users watching HTTP video-streaming videos are playback stallings, i.e., the events when the player stops the playback. One or two stalling events are enough to heavily impact the experience of the end user. Given that the analyzed measurements report the average per flow download throughput as one of the monitoring KPIs, we rely on our previous results to better understand how download throughput relates to QoE and stallings in YouTube.



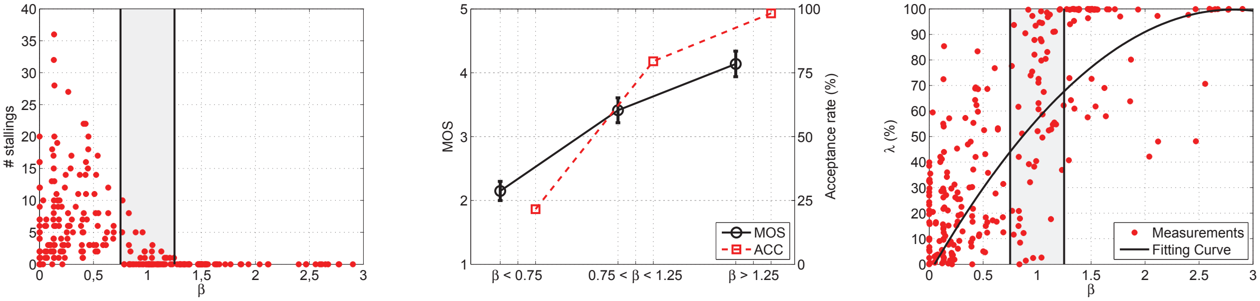
(a) YouTube overall QoE vs. downlink rate. (b) YouTube acceptability vs. downlink rate.

Figure 1. YouTube overall QoE and acceptability in terms of average downlink rate. The curves correspond to a best-case scenario, in which only 360p videos were considered. In a more general case with higher resolution videos (e.g., 1080p HD), the downlink rate has an even stronger effect on the user experience. The Figs. are taken from the study performed at [7].

A. QoE-based YouTube Monitoring

Fig. 1 reports the overall QoE and the acceptance rate as declared by users watching YouTube videos during a field trial test conducted and reported in [7], both as a function of the average download rate. During this one-month long field trial test, about 40 users regularly reported their experience on surfing their preferred YouTube videos under changing network conditions, artificially modified through traffic shaping at the core of the network. Fig. 1(a) shows the overall QoE as a function of the average download rate, using a 5-points MOS scale, where 1 corresponds to very bad QoE and 5 to optimal. The figure clearly shows that the overall QoE drops from a MOS score close to 4 at 800 kbps to a MOS score below 3 at 470 kbps. A MOS score of 4 corresponds to good QoE, whereas a MOS score below 3 already represents poor quality. The same happens with the service acceptance rate, as reported in Fig. 1(b). In the analysis, we shall consider the thresholds $T_{h_1} = 400$ kbps and $T_{h_2} = 800$ kbps as the throughput values splitting by bad, fair, and good QoE. Both curves correspond to a best-case scenario, in which only 360p videos were watched by the users. As we see next, both 360p videos and videos with higher resolutions are present in the dataset, thus QoE degradations are potentially worse than those reported.

In addition, we introduce a simple yet effective QoE-based KPI to monitor the QoE of YouTube videos from network measurements. In [8] we have already devised an approach to estimate stallings in YouTube from passive measurements at the core network, but the used techniques can not be applied when YouTube flows are carried over HTTPS, as it is currently happening. Therefore, using the same measurements of the field trial, we introduce a new approach. Intuitively, when the average download throughput (ADT) is lower than the corresponding video bit rate (VBR), the player buffer becomes gradually empty, ultimately leading to the stalling of the playback. We define β = ADT/VBR as a metric reflecting QoE. Fig. 2 reports (a) the measured number of stallings events and (b) the QoE user feedbacks as a function of β. In particular, no stallings are observed for $\beta > 1.25$, and user experience is rather optimal (MOS > 4). As a direct application of these results, if we consider standard 360p YouTube videos, which have an average VBR = 600 kbps [5], an ADT = 750 kbps would result in a rather high user QoE, which is the value



(a) # stallings vs. β. (b) MOS and acceptance vs. β. (c) User engagement (λ) vs. β.

Figure 2. β = ADT/VBR as a metric reflecting user experience and engagement. Users have a much better experience and watch videos for longer time when $\beta > 1.25$, corresponding to ADT = 750 kbps in 360p videos.

AS	**# IPs**	**#/24**	**#/16**	**% bytes**	**% flows**
All server IPs	3646	97	22	100	100
15169 (Google)	2272	60	2	80.8	77.3
43515 (YouTube)	1222	12	1	19.1	22.5
36040 (YouTube)	43	2	2	< 0.1	< 0.2

Table I. Number of IPs hosting YouTube, and shares of flows and bytes per AS.

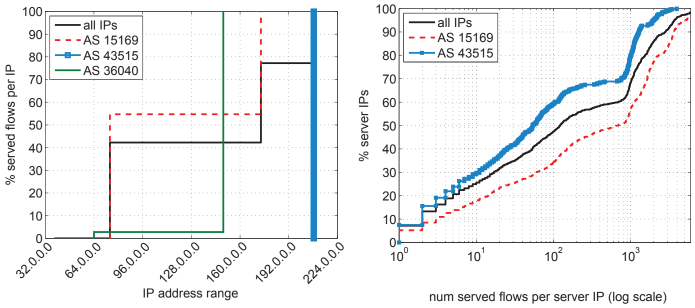
recommended by video providers in case of 360p videos. Fig. 2(c) additionally shows how the fraction λ = VPT/VD (video played time and duration) of the video time actually viewed by the end users actually increases when β increases, specially above the β = 1.25 threshold.

B. Understanding the YouTube Traffic

Before reporting the results of the YouTube performance degradation analysis, and in order to improve the understanding of the diagnosis process, we provide next an extensive characterization of the behavior of YouTube as observed in the first 4 days of the dataset. During these days we do not observe an important performance degradation, so therefore take the analysis as a reference of normal operation. The analysis considers the complete end-to-end service, describing (i) the hosting infrastructure, (ii) the traffic characteristics, and (iii) the performance of video delivery in terms of download flow throughput.

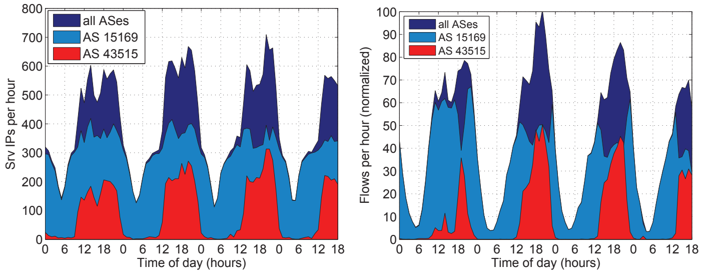
YouTube Hosting Infrastructure: Table I reports the number of unique server IPs serving YouTube, as well as the ASes holding the major shares of servers. To understand how these IPs are grouped, the table additionally shows the number of IPs per different network prefix. Two Google ASes hold the majority of the IPs (i.e., AS 15169 and AS 43515), grouped in a small number of /16 subnets. About 80% of the YouTube volume and number of flows are served by the AS 15169, whereas servers in AS 43515 are used for complementing the videos delivery to the customers of the monitored network.

To appreciate which of the aforementioned IP blocks host the majority of the YouTube flows, Fig. 3(a) depicts the distribution of the IP ranges and the flows per server IP. The majority of the YouTube flows are served by three well separated /16 blocks. Fig. 3(b) additionally depicts the number of flows served per server IP. Separated steps on the distributions evidences the presence of preferred IPs or caches serving a big number of flows, which are most probably selected by their low latency towards the end customers.



(a) Flows per server IP ranges. (b) Flows per server IP.

Figure 3. IP ranges and flows per server IP hosting YouTube. The majority of the YouTube flows are server by very localized IP blocks.

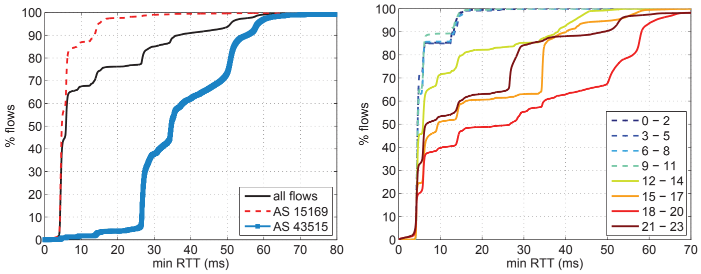


(a) IPs per hour hosting YouTube. (b) Flow counts per hour.

Figure 4. IPs and flows per hour. As much as 700 different IPs actively serve YouTube flows during peak-load hours.

Fig. 4 shows the dynamics of the traffic provisioning from the aforementioned IPs and ASes. Fig. 4(a) depicts the number of active IPs and Fig. 4(b) the flow counts per hour (normalized) for multiple consecutive days. As much as 700 different IPs actively serve YouTube flows during peak-load hours. Active IPs from either AS 43515 or AS 15169 show an abrupt increase at specific times of the day; for example, about 200 IPs from AS 43515 become active daily at about 10:00. In terms of flow counts, Fig. 4(b) evidences a very spiky behavior in the flows served from AS 43515, and some of the load balancing policies followed by Google, e.g., a drastic switch from AS 15169 to AS 43515 of the flows served at about 18:00.

How Far are YouTube Videos?: Google redirects user



(a) min RTT per AS.

(b) min RTT – daily variation.

Figure 5. min RTT to servers in different ASes. The server selection strategies performed by Google are not only based on closest servers.

requests to the closest server hosting the content in terms of latency [9]. Similar to [13], we investigate now the latency and the location of the previously identified servers, considering the distance to the vantage point in terms of Round Trip Time (RTT). The RTT to any specific IP address consists of both the propagation delay and the processing delay, both at destination as well as at every intermediate node. Given a large number of RTT samples to a specific IP address, the minimum RTT values are an approximated measure of the propagation delay, which is directly related to the location of the underlying server. It follows immediately that IPs exposing similar min RTT are likely to be located at a similar distance from the vantage point, whereas IPs with very different min RTTs are located in different locations. RTT measurements are passively performed on top of the YouTube flows.

Fig. 5 shows the distribution of the min RTT values for the flows observed in the analyzed 4 days. Steps in the CDF suggest the presence of different data-centers or clusters of co-located servers. Fig. 5(a) shows that about 65% of the flows come from servers most probably located in the same country of the ISP, as min RTT $<$ 5 ms. This is coherent with the fact that Google selects the servers with lower latency to the clients. A further differentiation by AS reveals that the most used servers in AS 15169 are located much closer than the most used servers in AS 43515. Fig. 5(b) depicts the dynamic behavior of the servers' selection and load balancing strategies used by Google to choose the servers. In particular, the figure reports the variation of the distribution of min RTT measured on the YouTube flows for a complete day, considering contiguous time bins of 3 hours length. Correlating these results with those in Fig. 4 permits to better understand the daily variations. Whereas the majority of the flows are served from very close servers until mid-day, mainly corresponding to AS 15169, servers in farther locations are additionally selected from 14:00 on, corresponding to the increase in the number of flows served from AS 43515.

YouTube Traffic and Performance: We study now the characteristics of the YouTube flows, as well as the performance achieved in terms of download throughput. Flows and video sizes, durations, and formats actually determine to a large extent the impact of the download throughput on the user experience, thus the interest of this analysis. Fig. 6 depicts the distribution of flow size for the different hosting ASes. Fig. 6(a) shows that about 20% of the flows are smaller than 1 MB. The CDF reveals a set of marked steps at specific flow sizes, for example at 1.8 MB and 2.5 MB. YouTube currently delivers 240p and 360p videos in chunks of exactly these sizes, explaining such steps. A similar behavior is observed for chunks of bigger sizes. About 75% of the flows are smaller than 4 MB, 90% of the flows are smaller than 10 MB, and a very small fraction of flows are elephant flows, with sizes higher than 100 MB. Fig. 6(b) depicts the distribution of the flows duration, in minutes. The flow duration is below 3 minutes for about 95% of the total flows. The abrupt step in the CDF at about 30 seconds is most probably linked to the aforementioned video chunk sizes, but we were not able to verify this observation. About 85% of the flows are shorter than 90 seconds. Fig. 6(c) shows the distribution of the video bitrate values. Almost 97% of the observed videos have a video bitrate smaller than 1Mbps, and the steps in the CDF at around 300kbps, 550kbps, and 800kbps correspond to the most preferred YouTube video formats present in our traces. To complement this picture, Fig. 6(d) shows the distribution of the video format, in terms of the YouTube itag values. The itag is an undocumented code used internally by YouTube to identify video formats (i.e., type and resolution). The largest majority of videos have itag codes 18, 22, and 34, corresponding to MP4 360p, MP4 720p, and FLV 360p video formats respectively.

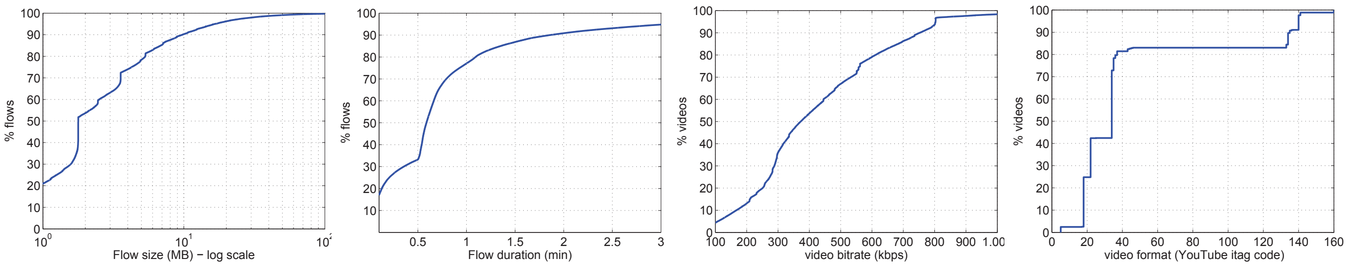
To conclude the characterization, Fig. 7 reports the distribution of the average download throughput. The figure consider only flows bigger than 1 MB, to provide more reliable and stable results (i.e., avoid spurious variations due to the TCP protocol start-up). More than 30% of the flows achieve a download throughput higher than 1 Mbps, whereas more than 15% of the flows achieve a throughput above 2 Mbps. Comparing figs. 7 and 6(c) it is rather difficult to understand whether the users are experiencing a proper QoE. Our manual inspection of the traces suggest that no major impairments were observed during this 4 day period. In the next section, we shall additionally show the analysis of the QoE-based KPI β to further understand how good is the QoE of the YouTube users in this network.

V. YouTube Anomaly Analysis

In this section we focus on the detection and diagnosis of the Google's CDN server selection policy negatively impacting the watching experience of YouTube users during several days at peak-load times. Conversations with the ISP confirmed that the effect was indeed negatively perceived by the customers, which triggered a complete Root Cause Analysis (RCA) procedure to identify the origins of the problem. As the issue was caused by an unexpected cache selection done by Google (at least according to our diagnosis analysis), ISP's internal RCA did not identify any problems inside its boundaries. As reported by the ISP operations team, the anomaly occurs on Wednesday the 8th of May. We therefore focus the analysis on the week spanning the anomaly, from Monday the 6th till Sunday the 12th. In the following analysis, we generally use 50% percentile values instead of averages, to filter out outlying values.

A. *Detecting the QoE-based Anomaly*

Fig. 8 plots the time series of three different performance indicators related to the YouTube download performance and to the end-user QoE. Fig. 8(a) depicts the median across all YouTube flows of the download flow throughput during the complete week. There is a normal reduction of the throughput



(a) YouTube flow size.
(b) YouTube flow duration.
(c) YouTube video bitrate.
(d) YouTube video format.

Figure 6. YouTube flows and video characteristics. Steps in the CDF in Fig.(a) at flow sizes 1.8 MB, 2.5 MB, 3.7 MB, etc. correspond to fixed chunk-sizes used by YouTube to deliver different video resolutions and bitrates. The largest majority of videos correspond to MP4 360p, MP4 720p, and FLV 360p formats.

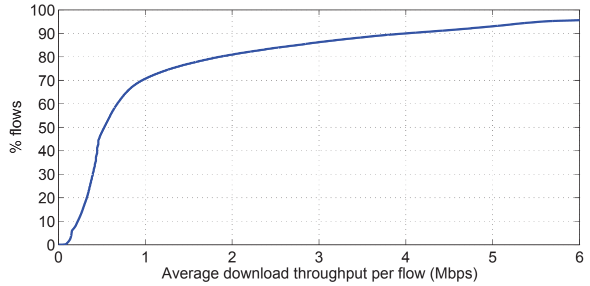
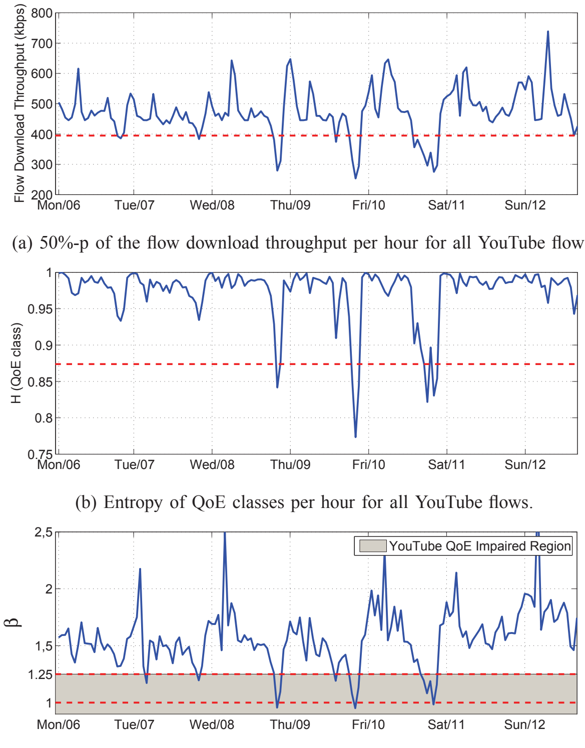


Figure 7. Average YouTube flow downlink throughput. More than 30% of the flows achieve a download throughput higher than 1 Mbps. The observed video bitrates suggest that the throughput is partially governed by the specific video bitrate and not exclusively by the network.

on Monday and Tuesday at peak-load time, between 20hs and 23hs. However, from Wednesday on, this drop is significantly higher, and drops way below the bad QoE threshold $T_{h_1} = 400$ kbps, flagging a potential QoE impact to the users. Fig. 8(b) plots the entropy of the QoE classes built from thresholds $T_{h_1} = 400$ kbps and $T_{h_2} = 800$ kbps, consisting of bad QoE for flows with average download throughput below T_{h_1}, fair QoE for flows with average download throughput between T_{h_1} and T_{h_2}, and good QoE for flows with average download throughput above T_{h_2}. Recall that these thresholds correspond to the QoE mappings presented in Fig. 1, which only cover 360p videos. Still, as depicted in Fig.6(d), the largest majority of the videos observed in the dataset corresponds to 360p videos and higher bitrate videos, thus T_{h_1} and T_{h_2} are somehow conservative thresholds, and QoE impairments might be even higher under the proposed QoE classes. The drop in the throughput combined with the marked drop in the time series of the QoE classes entropy actually reveals that a major share of the YouTube videos are falling into the bad QoE class. Finally, Fig. 8(c) actually confirms that these drops are heavily affecting the user experience, as the time series of the KPI β falls well into the video stallings region, depicted in Fig. 2.

B. Anomaly Diagnosis

The root causes of the detected anomalies can be multiple: the Google CDN server selection strategies might be choosing wrong servers, the YouTube servers might be overloaded, path changes with much higher RTT from servers to the customers might have occurred [16], paths might be congested, or there might be problems at the access network. Diagnosing problems at the access network is straightforward for the ISP, as this network belongs to itself. However, diagnosing the problem outside its boundaries is a much more complex task. As we said before, the ISP internal RCA did not identify any problems inside its boundaries, so we focus on the YouTube servers and on the download paths.



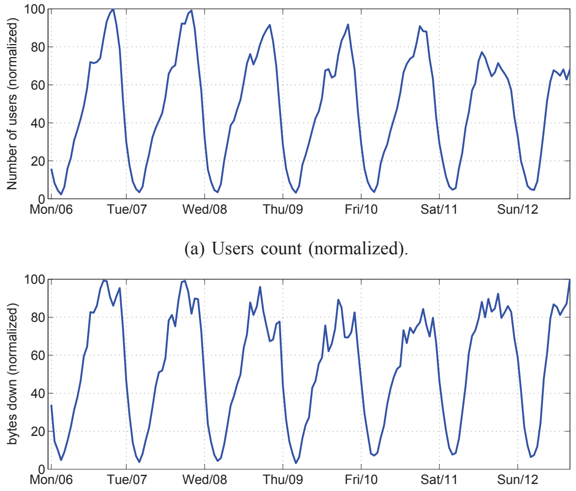
(a) 50%-p of the flow download throughput per hour for all YouTube flows.

(b) Entropy of QoE classes per hour for all YouTube flows.

(c) 50%-p of β per hour for all YouTube flows.

Figure 8. Detecting the QoE-based anomaly. There is a clear drop in the download flow throughput from Wednesday till Friday at peak-load hours, between 20hs and 23hs. The combined drop in the entropy of the QoE classes and in the KPI β reveal a significant QoE degradation.

Fig. 9 depicts the time series of the per hour users and bytes down normalized counts during the analyzed week. While there is a drop in the number of bytes down from Wednesday afternoon on, there are no significant variations on the number of users during the working week (i.e., Monday till Friday), so we can be sure that the throughput and QoE strong variations observed in Fig. 8 are not tied to statistical variations of the sample size. Using the results in Fig.2(c), we can say that the drop in the bytes down suggests that the bad QoE affected the users engagement with the video playing, resulting in users



(a) Users count (normalized).

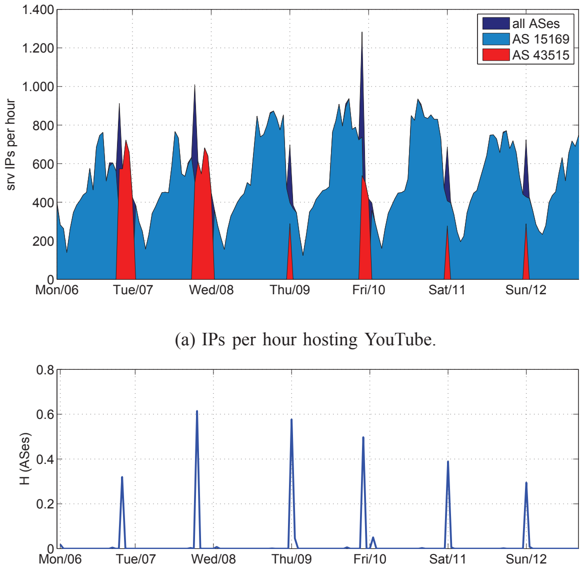
(b) Bytes down count (normalized).

Figure 9. Users and bytes down during the week of the anomaly. There are no significant changes during the specific times of the flagged anomaly.

dropping the watched videos when multiple stallings occur (i.e., when $\beta < 1.25$).

We study now the YouTube server selection strategy and the servers providing the videos. Fig. 10(a) depicts the number of server IPs providing YouTube flows per hour, similar to Fig.4(a). The first interesting observation is that the server selection policy used in the first 4 days of the dataset (15.04 - 18.04) and during the first 2 days of the week under study (06.05 - 07.05) is markedly different, specially in terms of servers selected from AS 43515. As depicted in Fig. 10(b), where the entropy of the AS number of the monitored server IPs is presented, there is a sharp shift of servers from AS 15169 to AS 43515 around peak-loud hours. In addition, there is an important reduction on the number of servers selected from AS 43515 on the days of the anomaly. This suggests that a different server selection policy is set up exactly on the same days when the anomalies occur.

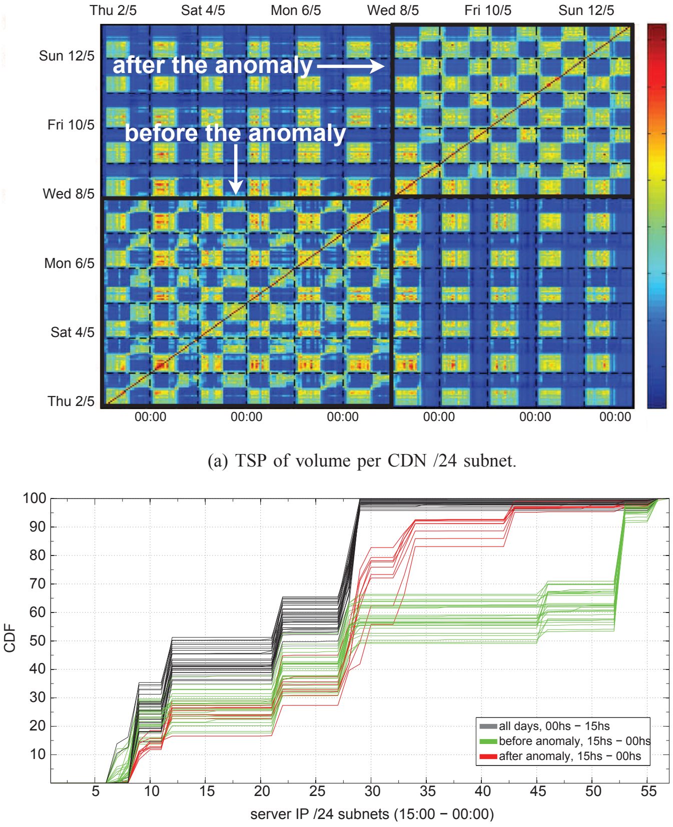
To further investigate this CDN server selection policy change, Fig. 11(a) shows the TSP of the video volume served by the different IPs in the dataset per hour, aggregated in /24 subnetworks, for 11 consecutive days. Recall that in the TSP, each point $\{i, j\}$ represents the degree of similarity between the distributions at hours t_i and t_j. The blue palette represents low similarity values, while reddish colors correspond to high similarity values. The TSP is symmetric around the 45° diagonal, thus the plot can be read either by column or by row. For a generic value of the ordinate at t_j, the points on the left (right) of the diagonal represent the degree of similarity between the past (future) distributions w.r.t. the reference distribution at t_j. Note the regular "tile-wise" texture within a period of 24 hours, due to a clear daily periodicity behavior in the selected servers. Specifically, there are two subnet sets periodically re-used in the first and second half of the day. The TSP clearly reveals that a different subnet set is used during the second half of the day from the 8th of May on, revealing a different cache selection policy. This change is also visible in the CDFs of the per subnet volume depicted in Fig. 11(b). Indeed, we can see that the same set of subnets is used between 00:00 and 15:00 before and after the anomaly, whereas the set used between 15:00 and 00:00 changes after the 8th, when the anomaly occurs.



(a) IPs per hour hosting YouTube.

(b) Entropy of the AS numbers time-series.

Figure 10. IPs hosting YouTube during the week of the anomaly.

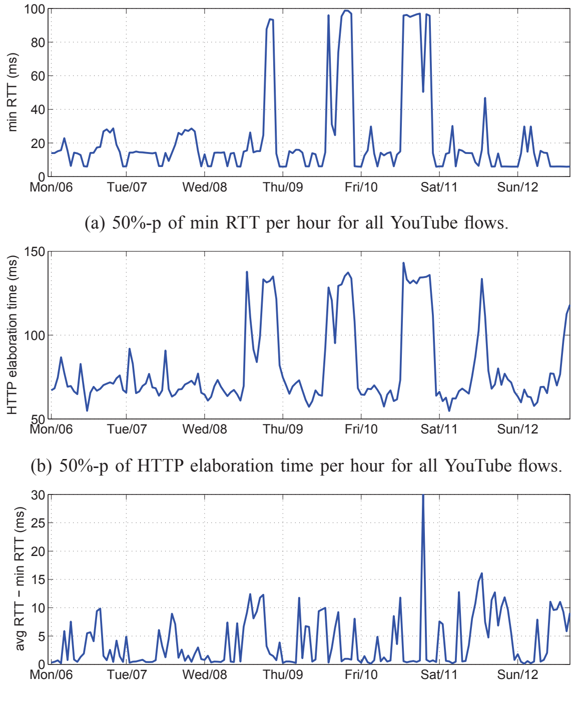


(a) TSP of volume per CDN /24 subnet.

(b) Volume per CDN /24 subnet.

Figure 11. Traffic volume distributions per CDN /24 subnets. There is a clear shift on the selected caches serving YouTube before and after the reported anomaly on Wednesday the 8th of May, specifically in the afternoon, between 15:00 and 00:00.

Given this change in the server selection policy, we try to find out if the problem arises from the newly selected servers, or if the problem is located in the path connecting these servers to the users. Fig. 12 studies the latency from users to servers



(a) 50%-p of min RTT per hour for all YouTube flows.

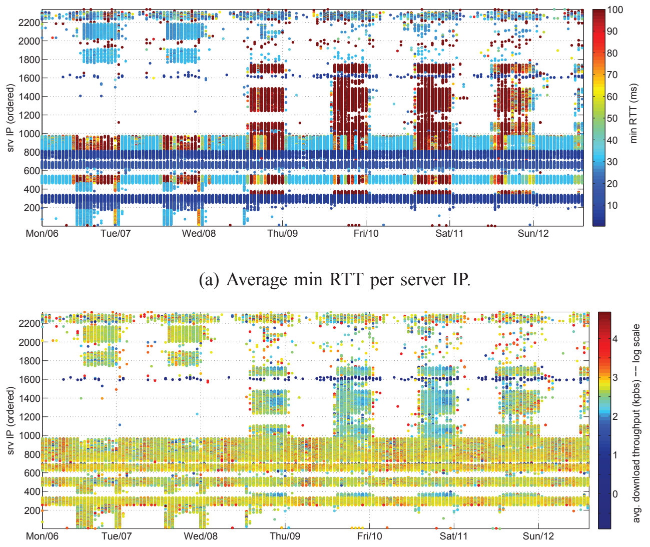
(b) 50%-p of HTTP elaboration time per hour for all YouTube flows.

(c) 50%-p of avg RTT - min RTT per hour for all YouTube flows.

Figure 12. The servers selected during the anomaly are much farther than those used before. While there is a marked increase in the server elaboration time, the difference between avg. and min. RTT remains bounded during the anomaly, so we discard the hypothesis of path congestion.

during the complete week. Fig. 12(a) depicts the median of the min RTT per hour as measured on top of all the YouTube flows. The marked increase in the RTT evidences that the servers selected during the anomaly are much farther than those used before the anomaly. This increase impacts directly on the HTTP elaboration time (i.e., time between HTTP request and reply), as depicted in Fig. 12(b). To understand if these latency increases are additionally caused by path congestion, Fig. 12(c) plots the time series of the difference between the min RTT and the average RTT values; in a nutshell, in case of strong path congestion, the average RTT shall increase (queuing delay), whereas the min RTT normally keeps constant, as it is directly mapped to the geo-propagation delay. The differences before and during the anomalies do not present significant changes, suggesting that the paths between servers and clients are not suffering from congestion. This is also confirmed by the analysis of the packet retransmissions, which do not present significant variations.

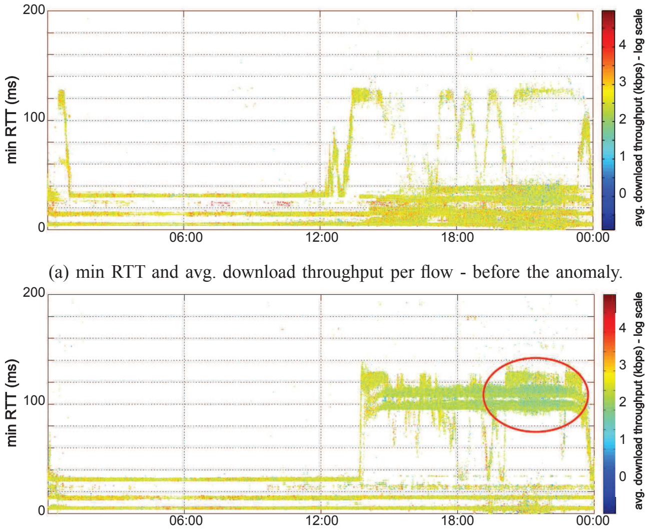
The last part of the diagnosis focuses on the YouTube servers. Fig. 13 depicts the average (a) min RTT and (b) download flow throughput per server IP in a heatmap like plot. Each row in the plots corresponds to a single server IP. The previously flagged min RTT increase is clearly visible for the new set of IPs which become active from 15:00 to 00:00 from Wednesday on. For those server IPs, Fig. 13(b) shows the important throughput drop during peak-load hours. Note however that large min RTT values do not necessary result in lower throughputs, as many of the servers used before and during the anomaly are far located but provide high throughputs. Fig. 14 further studies this drop, comparing the relation between min RTT and average download flow throughput before and during the anomaly. The increase of the min RTT is not the root cause of the anomaly. However, there is a clear cluster of low throughput flows coming from far servers during the peak-load hours.



(a) Average min RTT per server IP.

(b) Average download flow throughput per server IP.

Figure 13. There is a new set of server IPs providing YouTube videos from Wednesday on from farther locations. As visible in (b), the average download flow throughput for each of these new server IPs is much lower than the one obtained from other servers.

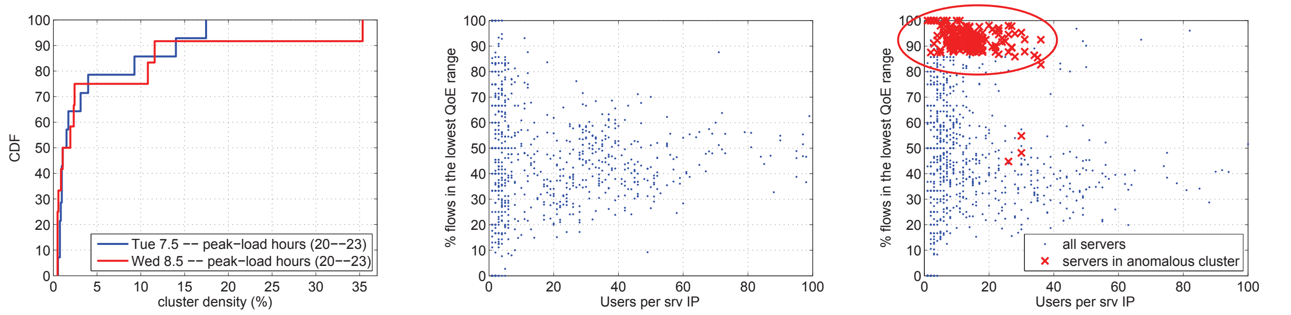


(a) min RTT and avg. download throughput per flow - before the anomaly.

(b) min RTT and avg. download throughput per flow - during the anomaly

Figure 14. The increase of the min RTT is not the root cause of the anomaly, as there are no major issues previous to the anomaly. However, there is a clear cluster of servers offering low throughput during the peak-load hours on an anomalous day.

The conclusion we draw from the diagnosis analysis is that the origin of the anomaly is the cache selection policy applied by Google from Wednesday on, and more specifically, that the additionally selected servers between 15:00 and 00:00 were not correctly dimensioned to handle the traffic load during peak hours, between 20:00 and 23:00. This shows that the dynamics of Google's server selection policies might result in poor end-user experience, on the one hand by choosing servers which might not be able to handle the load at specific times, or even



(a) Distribution of clusters density. (b) % bad QoE flows vs. # users, before anomaly. (c) % bad QoE flows vs. # users, before anomaly.

Figure 15. Unsupervised detection of the anomaly through clustering. There is a clear shift in the cluster density during the hours of the anomaly.

by selecting servers without considering the underlying end-to-end path performance.

C. Unsupervised Analysis

The last part of the paper briefly describes the unsupervised analysis of this kind of anomalies. The idea is to detect the occurrence of such events by tracking the evolution of the structure of the traffic, constructed through the DBSCAN clustering approach. In particular, we characterize each server providing YouTube traffic by a set of features used in the previous sections, including the number of flows, bytes, users, median download throughput, entropy of the QoE classes, fraction of flows in the lowest QoE class, and median of the previously studied latencies (i.e., min RTT, average RTT, and elaboration time), all of them computed in a temporal basis, i.e., per hour.

Fig. 15(a) depicts the distribution of the density of the clusters (measured in terms of fraction of server IPs contained in the cluster) identified during the peak-load hours, on a day previous to the anomaly and during the anomaly. There is a clear shift in the cluster density during the hours of the anomaly, revealing the appearance of a new cluster, containing about 35% of the servers. As presented in Figs. 15(b) and 15(c), the newly observed cluster corresponds to a set of server IPs providing a large share of YouTbe flows with low QoE, impacting a potentially large number of users. The interesting observation is that this set of server IPs can be identified by clustering, making it possible to detect the studied low performance events in an unsupervised manner.

VI. Concluding Remarks

In this paper, we have shown that the caching selection policies employed by a major CDN such as Google sometimes have an important impact on the end-customers QoE. Our results challenge OTT network performance evaluation approaches such as the Google's Video Quality Report, as these only highlight ISPs bandwidth provisioning as the only root cause of bad user experience. Through the analysis of one month of YouTube flow traces collected at the network of a large European ISP, we detected and drilled down a Google's CDN server selection policy negatively impacting the watching experience of YouTube users during several days at peak load times. We additionally presented different approaches to support the diagnosis, relying on YouTube QoE-based KPIs, time-series analysis, entropy-based approaches, and clustering techniques. Our work also presented a large-scale characterization of the YouTube service in terms of traffic characteristics and provisioning behavior of the Google CDN servers, useful to understand the normal and complex operation of YouTube. In the light of the emergence of new large-scale initiatives to measure the performance of ISPs delivering CDNs-based traffic, such as the Google's Video Quality Report, this paper offers explicit evidence showing that ISPs are not the only players responsible for poor end-user experience in Internet-scale services like YouTube.

References

[1] C. Labovitz et al., "Internet Inter-domain Traffic", in *ACM SIGCOMM*, 2010.

[2] V. Gehlen et al., "Uncovering the Big Players of the Web", in *TMA*, 2012.

[3] M. Zink et al., "Characteristics of YouTube Network Traffic at a Campus Network - Measurements, Models, and Implications", in *Computer Networks*, 2009.

[4] R. Torres et al., "Dissecting Video Server Selection Strategies in the YouTube CDN", in *IEEE ICDCS*, 2011.

[5] A. Finamore et al., "YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience", in *ACM IMC*, 2011.

[6] V. Menkovski et al., "Quality of Experience Models for Multimedia Streaming", in *Int. Journal of Mobile Computing & Multimedia Communications* 2(4), 2010.

[7] P. Casas et al., "YouTube & Facebook Quality of Experience in Mobile Broadband Networks", in *IEEE Globecom Workshops*, 2012.

[8] P. Casas et al., "YOUQMON: A System for On-line Monitoring of YouTube QoE in Operational 3G Networks", in *IFIP Performance*, 2013.

[9] R. Krishnan et al., "Moving Beyond End-to-End Path Information to Optimize CDN Performance", in *ACM IMC*, 2009.

[10] E. Nygren et al., "The Akamai Network: A Platform for High-Performance Internet Applications", in *SIGOPS* 44(3), 2010.

[11] M. Yu et al., "Tradeoffs in CDN Designs for Throughput Oriented Traffic", in *ACM CoNEXT*, 2012.

[12] A. D'Alconzo et al., "Distribution-based Anomaly Detection in 3G Mobile Networks: from Theory to Practice", in *IJNM* 20(5), 2010.

[13] P. Casas et al., "IP Mining: Extracting Knowledge from the Dynamics of the Internet Addressing Space", in *ITC 25*, 2013.

[14] J. Jiang et al., "Shedding Light on the Structure of Internet Video Quality Problems in the Wild", in *ACM CoNEXT*, 2013.

[15] P. Fiadino et al., "On the Detection of Network Traffic Anomalies in Content Delivery Network Services", in *ITC 26*, 2014.

[16] Y. Zhu et al., "LatLong: Diagnosing Wide-Area Latency Changes for CDNs", in *IEEE TNSM*, vol. 9(3), pp. 333–345, 2012.

[17] G. Nychis et al., "An Empirical Evaluation of Entropy-based Traffic Anomaly Detection", in *ACM IMC*, 2008.

[18] A. Finamore et al., "Experiences of Internet Traffic Monitoring with Tstat", in *IEEE Network*, vol. 25(3), 2011.

[19] A. Bär et al., "Large-Scale Network Traffic Monitoring with DBStream, a System for Rolling Big Data Analysis", in *IEEE BigData*, 2014.

[20] M- Ester et al., "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", in *ACM SIGKDD*, 1996.

[21] P. Casas et al., "UNADA: Unsupervised Network Anomaly Detection using Sub-Space Outliers Ranking", in *IFIP Networking*, 2011.