When YouTube doesn't Work – Analysis of QoE-relevant Degradation in Google CDN Traffic

P. Casas, A. D'Alconzo, P. Fiadino, A. Bär
FTW - Telecommunications Research Center Vienna
{surname}@ftw.at

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

T. Zseby
Vienna University of Technology
tanja.zseby@tuwien.ac.at

***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, both to improve their watching experience as well as to reduce the load on the core of the network, using dynamic server selection strategies. However, we show that such a dynamic approach can actually have negative effects on the end-user Quality of Experience (QoE). Through the comprehensive analysis of one month of YouTube flow traces collected at the network of a large European ISP, we report a real case study in which YouTube QoE-relevant degradation affecting a large number of users occurs as a result of Google's server selection strategies. We present an iterative and structured process to detect, characterize, and diagnose QoE-relevant anomalies in CDN distributed services such as YouTube. The overall process uses statistical analysis methodologies to unveil the root causes behind automatically detected problems linked to the dynamics of CDNs' server selection strategies.**

***Keywords*—YouTube; Content Delivery Networks; Traffic Monitoring; Quality of Experience; Statistical Data Analysis.**

I. Introduction

YouTube is the most popular video streaming service in today's Internet, being responsible for more than 30% of the overall Internet traffic [14], [18]. 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) [17] makes the overall picture even more complicated for Internet Service Providers (ISPs), as video requests from users are served from different servers at different times.

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. These control policies are typically very dynamic, causing large fluctuations in the traffic 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.

In this paper we consider the problem of detecting and diagnosing QoE-relevant performance degradation events in YouTube's traffic, using ISP-based measurements. Through the analysis of one month of YouTube flow traces collected at the network of a large European ISP, we 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[2]. We present an iterative and structured process to detect, characterize, and diagnose QoE-relevant anomalies in CDN distributed services, through the particular example of YouTube. The overall process uses statistical analysis methodologies to unveil the root causes behind automatically detected problems linked to the dynamics of CDNs' server selection strategies.

The main contributions of the paper are threefold: (i) firstly, we provide a large-scale characterization of the YouTube service in terms of traffic characteristics and provisioning behavior of the Google CDN servers. (ii) Secondly, we introduce simple yet effective QoE-based Key Performance Indicators (KPIs) to monitor YouTube videos from the end-user perspective, relying exclusively on ISP-based measurements. (iii) Finally and most important, our analysis provides evidence of the occurrence of QoE-relevant anomalies in YouTube induced by CDN server selection policies, which are 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 when Google is the culprit.

Note that 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 an overview on the papers characterizing YouTube, and those focusing on the analysis performance degradation issues. In Section III we describe the dataset used in the study, and present the data analysis approach we use, consisting of time-series analysis, entropy-based analysis, statistical distribution-based analysis, and clustering. Section IV introduces the QoE-based KPIs for YouTube monitoring, and Section V presents a characterization of the end-to-end YouTube service as observed from the collected traces. The analysis and complete diagnosis of the performance degradation in YouTube is performed in Section VI, additionally discussing an

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

[2] Conversations with the ISP confirmed that customers negatively perceived such degradations.

elaborated and structured approach for the diagnosis of QoE-relevant issues. Last but not least, Section VII updates the QoE perspective of YouTube used in our study, opening the door for future work in the field of QoE-based monitoring in YouTube from network measurements. Finally, Section VIII concludes this paper.

This work is an extended and more complete version of a recently published paper [3]. In particular, this work introduces the notions of anomaly diagnosis rules and diagnosis graphs, studies the geo-localization of the faulty servers, applies new detection techniques to identify abrupt changes in YouTube-relevant features, and introduces novel results in the QoE-based analysis of the YouTube service, specially when considering YouTube Dynamic Adaptive Streaming (DASH) scenarios. These new contributions are highly relevant, not only in terms of monitoring and detecting QoE-relevant issues in current YouTube video distribution, but also regarding the systematic identification of the potential root causes behind them.

II. Related Work

The study of the Internet traffic and applications delivered by CDNs has gained important momentum in the last few years [14], [18]. In particular, several studies characterize CDN architectures and focus on the optimization of their performance, server location, throughput and latency [1], [17], [23], [24], [26]. For example, [17] combines active measurements with routing and traffic data to identify causes of persistent performance problems for some CDN clients, whereas [26] focuses on the optimization of CDNs for throughput-oriented applications such as video streaming. In [24], authors present a tool that predicts the effects of possible configuration and deployment changes in CDNs. A very recent work [1] studies the popular Netflix service from a content distribution point of view, and shows that Netflix actually employs a blend of data centers and CDNs for content distribution in the US. The complexity of current CDNs makes of their study a highly challenging subject. As an example, Microsoft research presented a comparative analysis of two very popular CDNs [15], Akamai and Limelight, but the paper had to be withdrawn because of the narrow scope of the considered metrics for the study, which resulted in wrong conclusions.

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 [13], [25], [28], 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 [11], [16], [27]. In our recent work [11] 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 [27], 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 ms. 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 [16] 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. The main criticism to this paper is that it does not actually drill down into the potential root causes of the detected issues, and merely reports the most significant features related to the identified quality impaired sessions. The specific case study we evaluate in this paper would flag a combined CDN, Autonomous System (AS), and ISP related problem if we would follow the approach in [16], which would lead to incomplete and partially wrong conclusions.

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 or *relative frequency* 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 provided by the empirical probabilities, 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 [8]. 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 [12]. 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 [2]. Finally, using the server IP addresses (from now on, we shall use the term IP to refer to an IP address) of the flows, the complete dataset is complemented with the name of the Autonomous Systems (ASes) hosting the content, extracted from the MaxMind GeoCity ASes databases[3].

B. Entropy-based Analysis

The sample entropy has been proposed for traffic analysis in multiple contexts [19], [22]. Due to the similarity of the traffic analysis contexts (i.e., anomaly detection in network traffic), we particularly follow the techniques presented in [19], additionally using the normalization approach presented

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

in [22]. In a nutshell, given an empirical distribution of a certain variable, its sample entropy captures in a single value a measure of dispersion or concentration of the feature. More precisely, the entropy of a random variable X is defined as follows:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_2(p(x_i)), \quad (1)$$

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 probabilities, as the ratio between the number of observations taking value x_i and the total number of observations S. Assuming discretization of the observed values (i.e., a simple histogram), $p(x_i) = n_i/S$, where n_i corresponds to the number of samples inside the i-th discretization bin. Similar to [22], we normalize the sample entropy to the maximum entropy $log_2(N)$, where N is the total number of bins.

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 Ω, the KL divergence is defined as [8]:

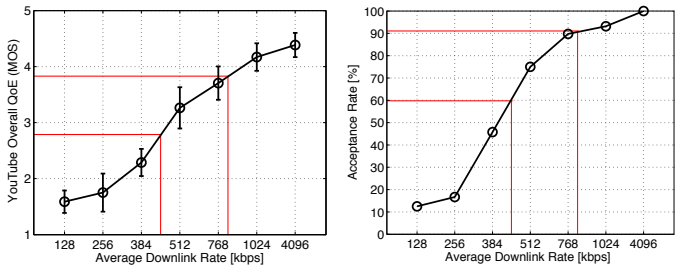
$$D(p||q) = \mathrm{E}\left[\log\left(\frac{p(\omega)}{q(\omega)}\right)\right] = \sum_{\omega\in\Omega} p(\omega)\log\left(\frac{p(\omega)}{q(\omega)}\right) \quad (2)$$

where the expectation is taken on $p(\omega)$, and following continuity arguments, $0\log\frac{0}{q} = 0$ and $p\log\frac{p}{0} = \infty$. The KL divergence provides a non-negative measure of the statistical divergence between p and q. It is zero $\leftrightarrow$ $p = q$, and for each $\omega \in \Omega$ it weights the discrepancies between p and q by $p(\omega)$. The KL divergence can not be actually considered as a distance metric, since it is not symmetric and does not satisfy the triangular inequality. Therefore, we adopted a more elaborated divergence metric, symmetric by construction:

$$L(p,q) = \frac{1}{2}\left(\frac{D(p||q)}{H_p} + \frac{D(q||p)}{H_q}\right) \quad (3)$$

where $D(\cdot||\cdot)$ is defined according to eq. (2), and H_p and H_q are the entropy of p and q respectively.

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 [9], 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 server selection policies used by Google to serve YouTube videos.



(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 figures are taken from the study performed at [6].

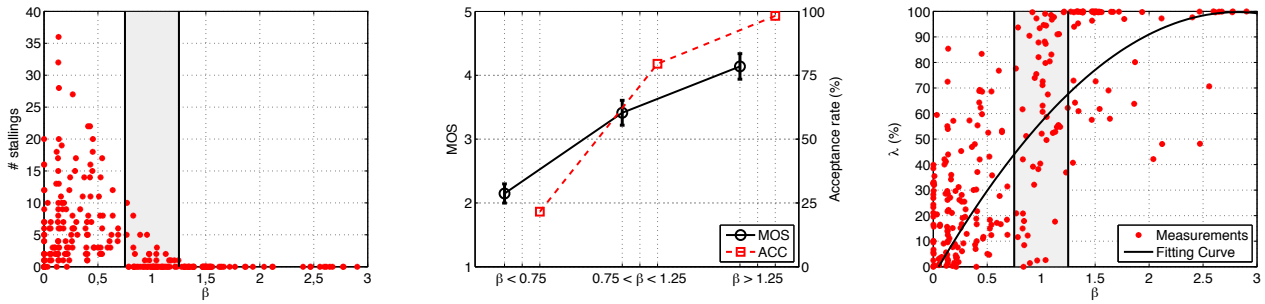
D. Distribution-based Analysis

The modified divergence metric can also be applied to detect abrupt changes in the empirical PDFs of relevant features. Therefore, using the results presented in [11], we shall use this divergence metric to detect anomalies in different features. In a nutshell, the proposed anomaly detection algorithm works by comparing the current probability distribution of a feature f to a set of reference distributions describing its "normal" behavior. Traditional approaches for network anomaly detection consider individual and independent time series analysis, processing different traffic descriptors or features with classical forecasting and outliers analysis methods. Using a probability distributon-based approach is intrinsically more powerful, as it considers the entire distribution of different traffic features, rather than only specific moments of the random variable distributions (e.g., mean-based, percentile-based, or variance-based change detection). The specific types of features we use in this work capture both the intrinsic and dynamic CDNs mechanisms (e.g., number of flows and bytes served by each CDN server IP), and end-users experienced performance (e.g., flow download throughput). A full description of this algorithm is presented in [9].

E. 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 [10], 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.

There are tens of well-known clustering algorithms in the literature, but our selection of DBSCAN has a clear motivation: 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 [5].



(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. The marked region $0.75 < \beta < 1.25$ represents the critical QoE-relevant point of operation, where QoE starts to heavily degrade. Users have a much better experience and watch videos for longer time when $\beta > 1.25$, corresponding to ADT = 750 kbps in 360p videos.

IV. Quality of Experience-based YouTube Monitoring

Even if the download throughput has a direct impact on the performance of YouTube provisioning [20], our previous studies [6], [7] 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.

Figure 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 [6], 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. Figure 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 Figure 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.

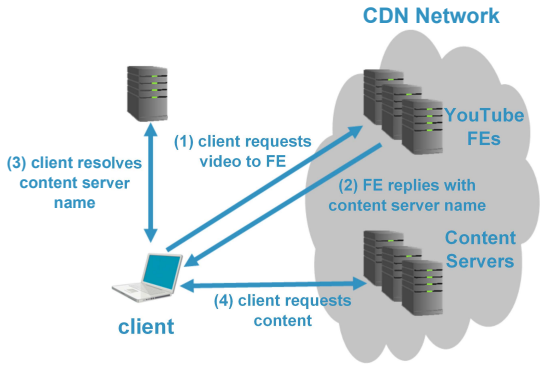
In addition, we introduce a simple yet effective QoE-based KPI to monitor the QoE of YouTube videos from network measurements. In [7] we have already devised a Deep Packet Inspection based approach to estimate stallings in YouTube from passive measurements at the core network. However, the used techniques can not be applied when YouTube flows are carried over HTTPS as it is currently happening, simply because it is no longer possible to access the encrypted content of the traffic. 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. Figure 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 [13], an ADT = 750 kbps would result in a rather high user QoE, which is the value recommended by video providers in case of 360p videos. Figure 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 $\beta = 1.25$ threshold.

Table I. Number of IPs hosting YouTube, and shares of flows and bytes per AS. AS 15169 hosts the *preferred* YouTube caches.

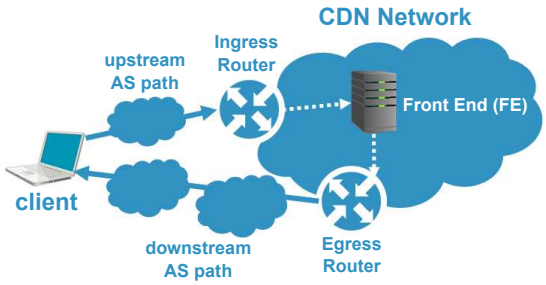
AS	**# IPs**	**#/24**	**#/16**	**% bytes**	**% flows**
All server IPs	3646	97	22	100	100
15169 (Google)	2272	60	2	80.8	77.3
43515 (YouTube)	1222	12	1	19.1	22.5
36040 (YouTube)	43	2	2	< 0.1	< 0.2

V. YouTube Traffic Characterization

YouTube replicates content across geo-distributed data-centers worldwide to improve the overall performance of the video content provisioning. Google's CDN uses a complex content location and server selection strategy for optimizing client-server latency, increase QoE in general, and perform load balancing. User requests are normally redirected to the closest servers, based on Round Trip Time (RTT) measurements. For doing so, YouTube keeps a periodically updated latency map between its servers and BGP prefixes aggregating geo co-located users [27]. As depicted in Figure 3(a), Google uses the DNS service for redirecting requests to the preferred servers, additionally using dynamic cache selection strategies to balance the load among YouTube servers. YouTube Front End (FE) servers are those handling the original user request for a specific video, which can then redirect the user to



(a) Video retrieval workflow (extended figure, original source at [25]).



(b) Google's CDN (extended figure, original source at [27]).

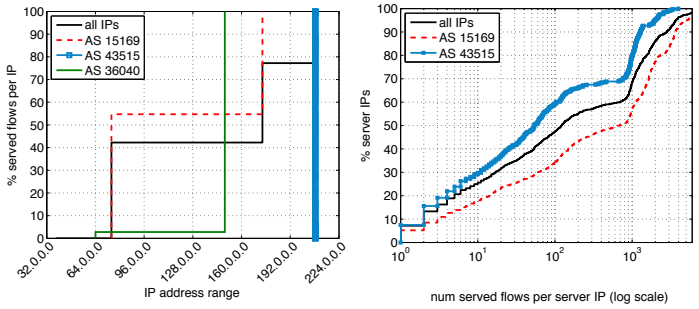
Figure 3. YouTube workflow for video retrieval and content location. Google's CDN uses a complex content location and server selection strategy for optimizing client-server latency, increase QoE in general, and perform load balancing. DNS is used for request re-directioning.

additional YouTube servers mirroring the content. In some cases, YouTube servers located at multiple ASes of distance are selected (see Figure 3(b)), resulting in higher delays and potentially impacting the performance of the video delivery in terms of download throughput.

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.

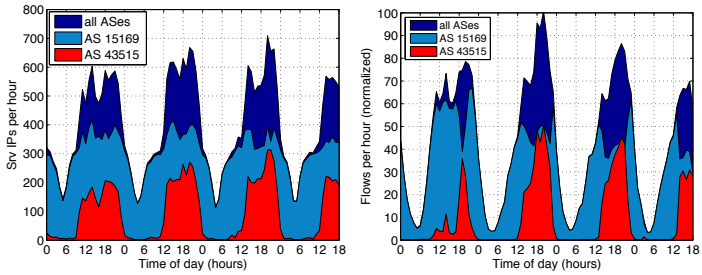
YouTube Hosting Infrastructure: Table I reports the number of unique server IPs serving YouTube, as well as the ASes holding the major shares of servers. To understand how these IPs are grouped, the table additionally shows the number of IPs per different network prefix. Two Google ASes hold the majority 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, Figure 4(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



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

Figure 4. IP ranges and flows per server IP hosting YouTube. The majority of the YouTube flows are server by very localized IP blocks.



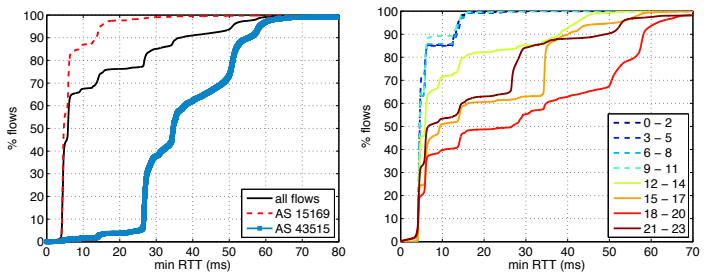
(a) IPs per hour hosting YouTube. (b) Flow counts per hour.

Figure 5. IPs and flows per hour. As much as 700 different IPs actively serve YouTube flows during peak-load hours.

separated /16 blocks. Figure 4(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.

Figure 5 shows the dynamics of the traffic provisioning from the aforementioned IPs and ASes. Figure 5(a) depicts the number of active IPs and Figure 5(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, Figure 5(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? As we said, Google redirects user requests to the closest server hosting the content in terms of latency [17]. Similar to [4], 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 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, 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



(a) min RTT per AS. (b) min RTT – daily variation.

Figure 6. min RTT to servers in different ASes. The server selection strategies performed by Google are not only based on closest servers.

located in different locations. RTT measurements are passively performed on top of the YouTube flows.

Figure 6 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. Figure 6(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. Figure 6(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 Figure 5 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. Figure 7 depicts the distribution of flow size for the different hosting ASes. Figure 7(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. Figure 7(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. Figure 7(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, Figure 7(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.

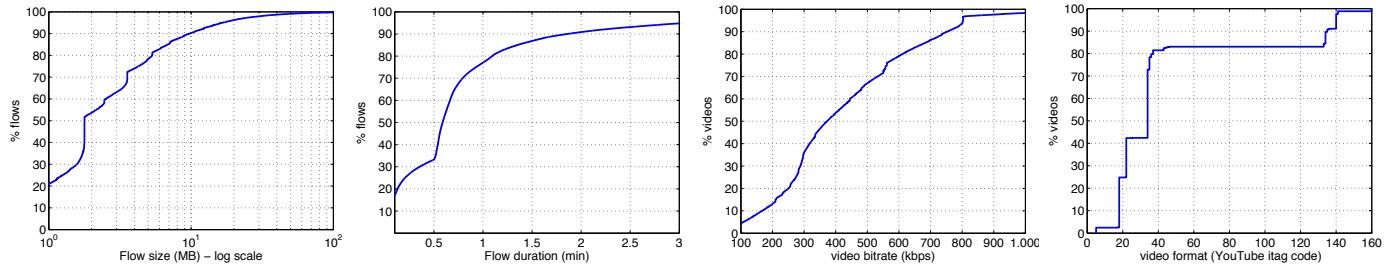
To conclude the characterization, Figure 8 reports the distribution of the average download throughput. The figure consider only flows bigger than 1 MB, to provide more reliable and stable results (i.e., avoid spurious variations due to the TCP protocol start-up). More than 30% of the flows achieve a download throughput higher than 1 Mbps, whereas more than 15% of the flows achieve a throughput above 2 Mbps. Comparing Figures 8 and 7(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.

VI. 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-relevant Anomaly

Figure 9 plots the time series of three different performance indicators related to the YouTube download performance and to the end-user QoE. Figure 9(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 on Monday and Tuesday at peak-load time, between 20:00 and 23:00 UTC. However, from Wednesday on, this drop is much larger, and drops way below the bad QoE threshold $T_{h_1} = 400$ kbps, flagging a potential QoE impact to the users. Figure 9(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 Figure 1, which only cover 360p videos. Still, as depicted in Figure 7(d), the largest



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

Figure 7. YouTube flows and video characteristics. Steps in the CDF in Figure (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.

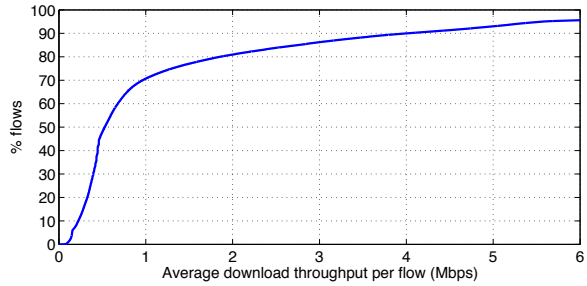
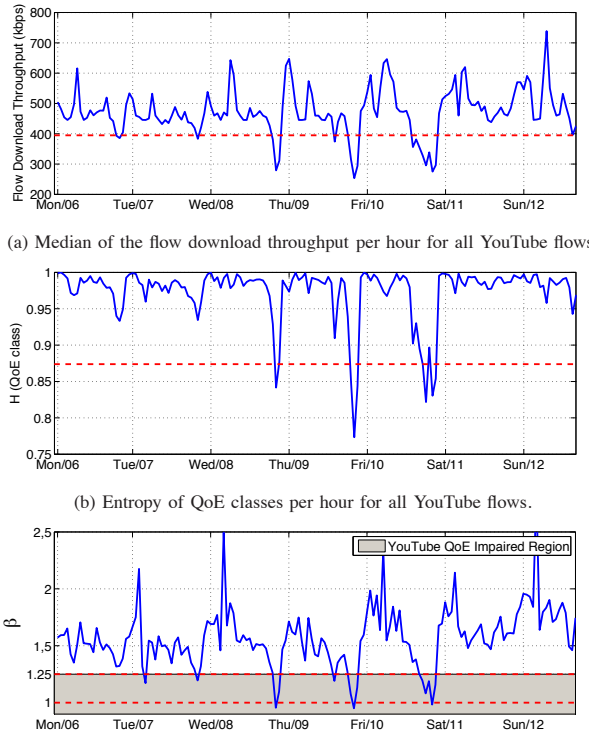


Figure 8. Average YouTube flow downlink throughput. More than 30% of the flows achieve a download throughput higher than 1 Mbps. The observed video bitrates suggest that the throughput is partially governed by the specific video bitrate and not exclusively by the network.

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, Figure 9(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 Figure 2.

The anomaly can also be statistically detected as a large deviation on the distribution of relevant features, for example, in the distribution of the average download flow throughput. Figure 10 reports the output of the distribution-based anomaly detection algorithm described in Section III-D. The algorithm computes the distance between the hourly-calculated empirical probabilities of the average download flow throughput, and flags an anomaly when this distance is higher than a certain confidence interval threshold. Figure 10(a) depicts the time series of the obtained KL-based divergences when comparing current distribution to a reference set of distributions considered as normal. The dotted lines represent the evolution of the confidence interval of the algorithm, and the red markers flag the detected anomalous time slots. The algorithm systematically raises alarms only during the peak hours (21:00-23:00) from the 8th onward, matching exactly the times of the QoE degradations flagged in Figure 9(c). Note how the algorithm does not flag anomalies on Saturday the 11th on peak hours, being consistent to Figure 9. Finally, Figure 10(b) depicts the distribution of the average video flows download rate at peak



(a) Median of the flow download throughput per hour for all YouTube flows.

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

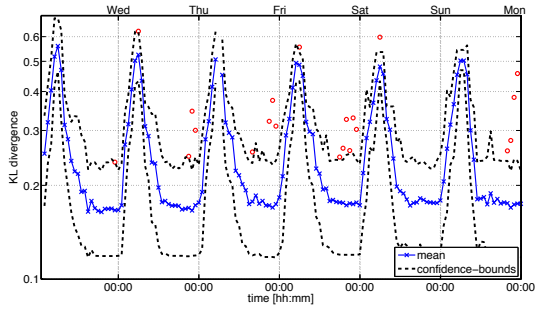
(c) Median of β per hour for all YouTube flows.

Figure 9. Detecting the QoE-relevant anomaly. There is a clear drop in the download flow throughput from Wednesday till Friday at peak-load hours, between 20:00 and 23:00 UTC. The combined drop in the entropy of the QoE classes and in the KPI β reveal a significant QoE degradation.

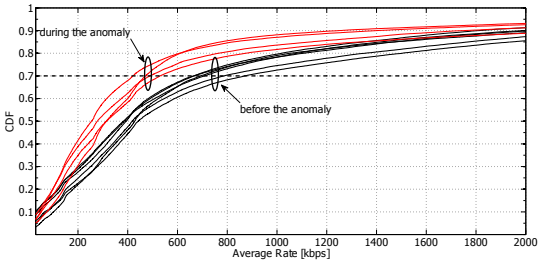
hours, both before and during the anomaly. There is a clear reduction on the video flows download throughput during the anomaly, which results in the aforementioned QoE-relevant impairments.

B. Anomaly Diagnosis Howto

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, paths might be congested, there might be problems at the access network or even at the end devices. Diagnosing problems at the access network is somehow easier for the ISP, as this network belongs to itself (even if in the general case it can still be a very challenging task for ISPs).



(a) Anomalies detected in the average download flow throughput. The red markers correspond to the flagged anomalies. The gap in KL divergence on Thursday morning is caused by maintenance of the Anomaly Detection tool.



(b) Throughput distributions before and during the anomaly.

Figure 10. Detection of anomalies in YouTube traffic. Alarms and acceptance region for the distribution of video flows average download rate.

However, diagnosing the problem outside its boundaries is a much more complex task.

The general approach for diagnosing network and traffic anomalies is to follow an iterative analysis of the possible root causes originating the problems. Such an iterative analysis is done in the practice by applying a set of diagnosis rules to verify the occurrence (or not) of specific signatures explaining the detected symptoms. These rules are initially defined by an expert operator, based on his domain knowledge and operational experience. Given a specific issue to diagnose – in this specific use case, an important QoE-degradation impacting a large number of users watching YouTube – each of the rules checks for a predefined signature characterizing the root causes.

To define the set of knowledge based rules to diagnose a problem, the first step is to identify which are the possible root causes of such problems, and where could the origins be located. The large number of possible root causes coupled with the generally much lower number of vantage points providing information about the symptoms makes the enumeration of the root causes and their location a complex task. The approach we take in this paper is a coarse one, in which we drill down the detected anomaly to find out the main part of the end-to-end service delivery responsible for it (e.g., device, access ISP, Internet, CDN, content provider), rather than the specific network element (e.g., interconnection router, link failure, routing table, etc.). In our specific case study, the origins of the QoE-relevant degradation could be potentially located at:

(i) end terminals: potential issues in the end-terminal are multiple, from software to hardware issues, as well as connectivity and signal strength among others. However, as we said before, this case study considers QoE impacts in a large number of users, and thus individual buggy terminal events are out of the scope of the diagnosis analysis. Only problems simultaneously affecting a large number of terminals are potentially considered; for example, issues related to software updates affecting a whole category of devices (i.e., iOS smartphones, Windows 8 OS, etc.).

(ii) home network: similar to previous observations for end terminal issues, the home network could be a potential issue only in case of problems affecting for example a whole category of home gateway devices. However, in this specific case, firmware updates are much less frequent than OS and software updates, and therefore we exclude the home network from the analysis.

(iii) access network: diagnosing issues at the access network heavily depends on the type of access network considered (cellular, WiFi, FTTH, ADSLx). Download throughput problems at the access can be caused by multiple issues, from congestion events to equipment outages and misconfigurations.

(iv) core network of the ISP: problems at the ISP providing the Internet access to the users are generally the most common ones. These are various, including intra-AS routing, router outages and equipment failures, misconfigurations, etc. The usage of virtualization and software-defined technologies (both the access and core networks) adds additional sources of potential performance issues.

(v) Internet: depending on the location of the YouTube content and on the cache selection policies used by Google to answer users' requests, the YouTube flows might have to traverse multiple ASes from the YouTube servers till reaching the access ISP. As we said before, YouTube would normally assign user requests to the closest servers. Still, due to its load balancing policies, YouTube might assign users to other servers farther located, resulting in multi-AS paths from servers to customers. As a consequence, problems related to inter-AS routing, congestion at intermediate ASes, and multi-AS paths performance degradation are potential root causes for YouTube QoE degradation.

(vi) CDN and the servers: the final part of the end-to-end service diagnosis corresponds to the servers hosting and providing the YouTube videos. Software or hardware problems of the hosting servers, overloading situations of wrongly dimensioned servers, internal problems of the hosting datacenter, etc. are possible root causes to additionally diagnose.

Once we have enumerated the list of elements to diagnose, we can define a set of rules or *check-list* which shall be iteratively verified to detect the occurrence of events revealing the aforementioned problems. Table II enumerates a non-exhaustive list of the domain-knowledge based rules for diagnosing the QoE-drop event detected in YouTube.

These diagnosis rules can be structured as a *diagnosis graph*, which is used for guiding the diagnosis and drill-down of the YouTube QoE-anomaly. Figure 11 depicts an exemplifying decision graph, integrating some of the previous diagnosis rules. The branches of a decision graph can be either conditionally or systematically followed. In our case, the analysis is conditional, starting from the end terminals till reaching the CDN servers.

The decision graph is structured in five different blocks: the *(1) QoE-relevant Anomaly Detection* block consists of

Table II. SET OF DIAGNOSIS RULES/ITEMS TO CHECK FOR DIAGNOSING PERFORMANCE ISSUES IN CDN SERVICES SUCH AS YOUTUBE.

Where?	Potential Root Cause and/or Location	Check-list Items – Diagnosis Rules
Terminals and Home Networks	**Device**	For all the involved user devices corresponding to the affected flows, check the occurrence of end-device issues.
	Device OS	For all the involved user devices corresponding to the affected flows, check the heavy hitters of OS type, and the entropy of the OS class.
	Set Top Box	For all the involved boxes corresponding to the affected flows, check the heavy hitters of box-type, and the entropy of the box-type class.
Access Network	**Access Overloading**	Check the occurrence of access-overloading events during the last days, for the corresponding access networks or logical aggregation points (e.g., users in the same aggregation network, or attached to the same DSLAM, etc.). Compare to similar events for other users accessing the same servers through a different access network.
	Access Configuration	Check the occurrence of reconfiguration events related to the corresponding access networks.
	Equipment Failure	Check the occurrence of outage events reported by the KPIs monitored by the ISP at the corresponding access networks.
Core Network	**Intra-AS Routing**	For all the involved user devices corresponding to the affected flows, check the occurrence of end-device issues.
	Link Congestion	Check co-occurrence of link congestion events.
	Equipment Failure	Check the occurrence of outage events reported by the KPIs monitored by the ISP on its internal equipment, including routing/switching/forwarding equipments.
Internet	**Inter-AS Routing**	Check end-to-end path change events in the corresponding temporal span of the detected anomaly.
	Path Congestion	Check flagged events related to abrupt increases in packet retransmissions per server, or in the end-to-end queuing delay, for all the flows provisioned by the corresponding servers.
	Intermediate AS Issues	Check performance degradation events in the intermediate ASes, particularly including latency and congestion in the different end-to-end ASes path segments.
CDN Servers	**Server Reachability**	Check if geo-distributed reachability measurements to the identified servers result in non-reachability problems.
	Server Soft/Hard Failure	Check occurrence of server hardware outages and/or software-related events at each single identified server IP during the time span of the detected anomaly.
	Server Overloading	Check occurrence of overloading events at each single identified server IP during the time span of the detected anomaly.

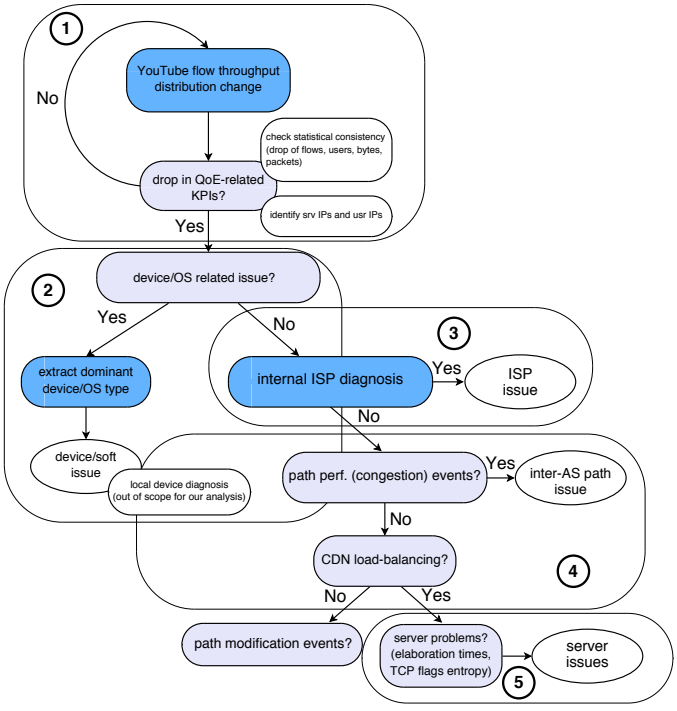
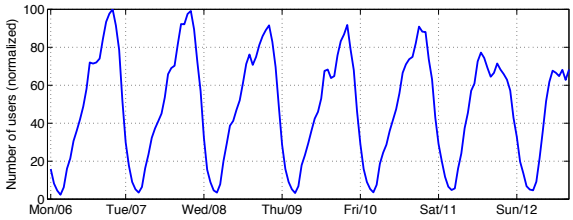


Figure 11. Diagnosis graph associated to the detection and troubleshooting support of large-scale QoE-relevant anomalies in YouTube.

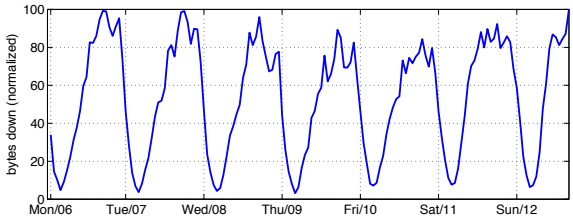
the anomaly detection approaches (both entropy-based and distribution-based), coupled with the QoE-based monitoring for understanding whether the detected changes are causing QoE-relevant degradations or not. To avoid triggering the complete diagnosis process on false alarms caused by statistical variations of the monitored features, this block additionally adds a verification of the consistency of the detected anomaly. For example, important deviations in the empirical distribution of the β KPI can be caused by a sudden and important drop/increase in the number of YouTube flows, or by an abrupt modification in the number of users watching YouTube. Therefore, the verification step firstly checks for the presence of events related to major statistical variations in the number of YouTube flows and the number of users watching YouTube. The consistency step additionally defines an hysteresis-based approach for triggering the diagnosis, in which a number of consecutive anomaly alarms have to be flagged before launching the drilling down process. The *(2) End-device Diagnosis* block focuses on the specific analysis of the type of end device associated to the anomalous YouTube flows. The *(3) ISP Diagnosis* block consists of the diagnosis of the access ISP. The *(4) Internet paths Diagnosis* block focuses on the diagnosis of the end-to-end inter-AS paths, including both routing and path congestion analysis. Finally, the *(5) CDN servers Diagnosis* block allows to identify server-related performance issues from end-to-end measurements, assuming that access to in-CDN measurements is not available. Note that these five blocks do not fully cover the aforementioned set of domain-knowledge based rules. Still, the description serves as an example on how to build a diagnosis graph to tackle the case study under analysis, which we do next.

C. The Diagnosis in the Practice

As we said before, in this case study we exclude potential problems at the end devices or home networks, as we are targeting a large-scale anomaly, impacting a large share of the monitored customers. In addition, we recall that the ISP internal RCA did not identify any problems inside its boundaries, so we also exclude the ISP network from the analysis.

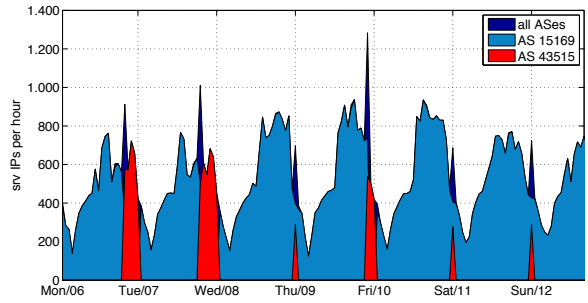


(a) Users count (normalized).

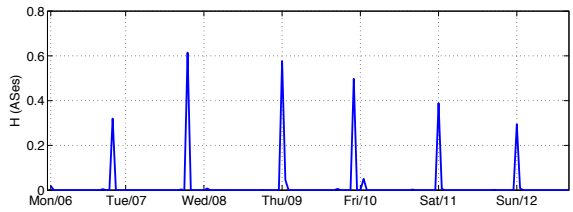


(b) Bytes down count (normalized).

Figure 12. Users and bytes down during the week of the anomaly. There are no significant changes during the specific times of the flagged anomaly.



(a) IPs per hour hosting YouTube.

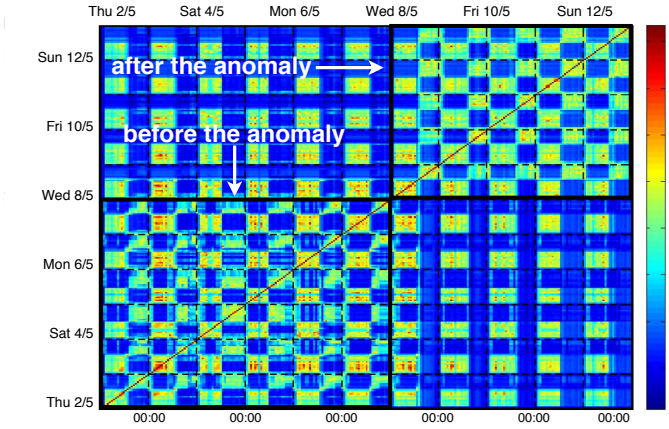


(b) Entropy of the AS numbers time-series.

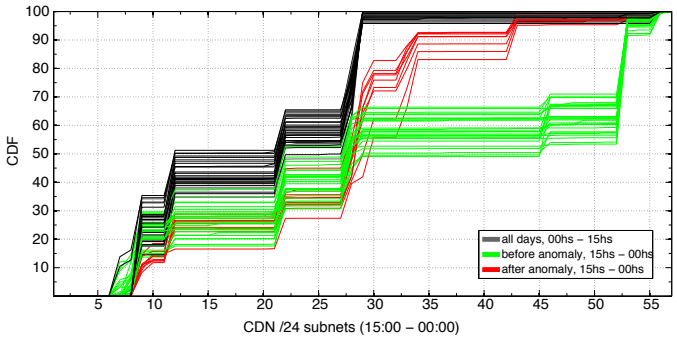
Figure 13. IPs hosting YouTube during the week of the anomaly.

Therefore, we shall only focus on the YouTube servers and on the download paths performance.

Figure 12 depicts the time series of the per hour users and bytes downloaded normalized counts during the analyzed week. While there is a drop in the number of bytes downloaded 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 Figure 9 are not tied to statistical variations of the sample size. Using the results in Figure 2(c), we can assume that the drop in the bytes downloaded suggests that the bad QoE affected the users engagement with the video playing, resulting in users dropping the watched videos when multiple stallings occur (i.e., when $\beta < 1.25$). Let us focus now on the YouTube server selection strategy and the servers providing the videos.



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

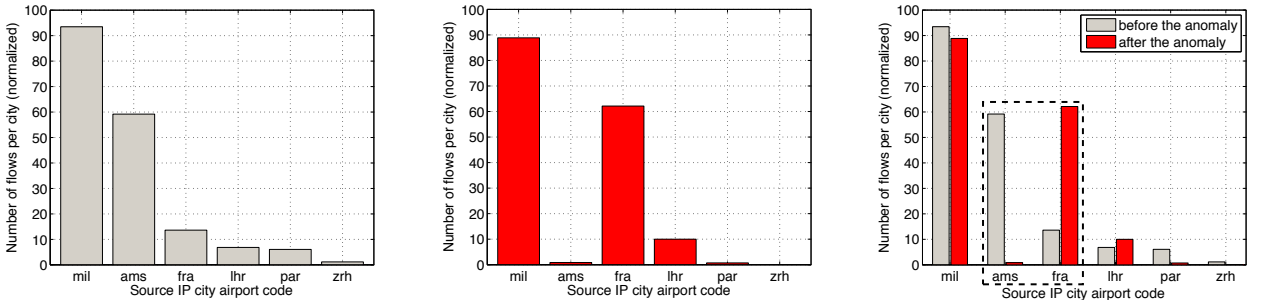


(b) Volume per CDN /24 subnet.

Figure 14. 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.

Figure 13(a) depicts the number of server IPs providing YouTube flows per hour, similar to Figure 5(a). The first interesting observation is that the server selection policy used in the first 4 days of the dataset (April the 15th till the 18th) and during the first 2 days of the week under study (May the 6th and the 7th) is markedly different, specially in terms of servers selected from AS 43515. As depicted in Figure 13(b), where the entropy time-series of the AS distribution corresponding to the monitored server IPs is presented, there is a sharp shift in the distribution of hosting ASes around peak-load hours. This shift corresponds to server IPs selected from AS 43515 rather than from AS 15169. 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, Figure 14(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



(a) Normal daily flows count per city. (b) Anomalous daily flows count per city. (c) Shift on the daily flows per top-6 cities.

Figure 15. Geo-localization of the detected anomaly. There is a major shift in the daily number of YouTube flows coming from servers in Amsterdam to Frankfurt, suggesting that the problem is linked either to servers in Frankfurt, or to the new server-to-customer network paths.

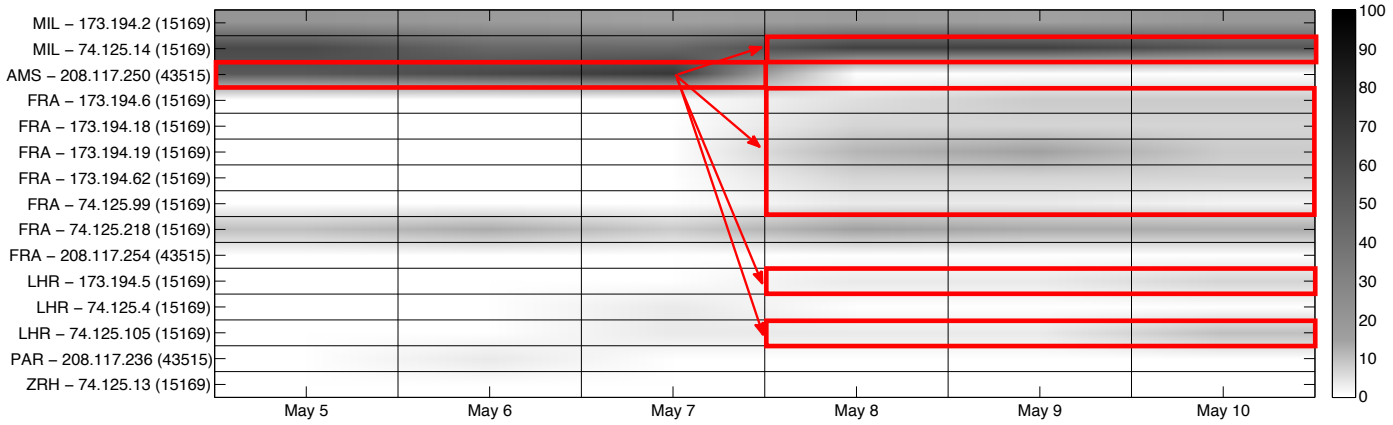


Figure 16. Daily distribution of the YouTube flows per city and /24 subnetwork. Each column adds to 100%, and the darker the color, the higher the fraction of flows hosted. Starting on May the 8th, the lion share of the YouTube flows, normally served from Amsterdam, are shifted to Frankfurt and London.

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 Figure 14(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.

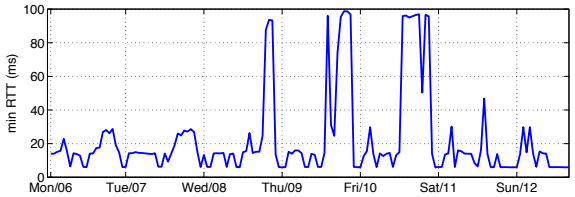
We take a step further in characterizing this CDN server selection policy, by taking a server geo-localization approach. The DNS-based re-directioning used by YouTube imposes a specific structure on the video identifiers requested to the content servers, which additionally include the name of the city where the server hosting the requested content is located. This city name is formated as an airport code, better known as IATA code (e.g., FRA for Frankfurt, AMS for Amsterdam, etc.). YouTube obfuscates this information, but it can be retrieved by reverse engineering (a description on how to do it is out of scope). Using this information, we can study the geographical location of the new servers selected from the 8th on.

Figure 15 reports the daily number of flows (normalized) served from the top cities hosting the YouTube content in our traces on (a) a day before the 8th of May and (b) a day after the 8th of May. The top cities hosting the YouTube videos in this case study are Milano and Amsterdam, followed by Frankfurt and other EU cities. The comparison presented in Figure 15(c) shows that the newly selected servers are mainly located in Frankfurt and London, and that almost all the flows served from Amsterdam are shifted to these cities in the new cache-selection policy. Figure 16 complements this geo-localization view on the traffic by reporting the daily distribution of the YouTube flows per city and per /24 subnetwork and AS. The shift is done from a single /24 subnetwork in AS 43515 to more than five /24 subnetworks in AS 15169. Very interestingly, the servers located in Amsterdam are almost no longer used after the shit on the 8th.

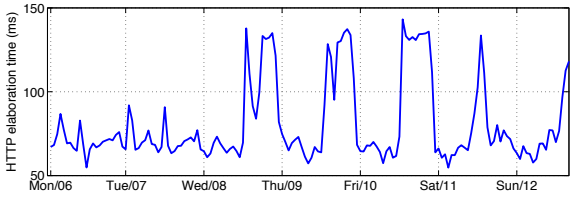
Given this change in the server selection policy, we try to find out if the problem arises from the newly selected servers, or if the problem is located in the path connecting these servers to the users. Figure 17 studies the latency from users to servers during the complete week. Figure 17(a) depicts 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

	May 5	May 6	May 7	May 8	May 9	May 10
MIL – 173.194.2 (15169)	1330	1275	1250	1300	1200	1350
MIL – 74.125.14 (15169)	1250	1250	1160	1300	1220	1140
AMS – 208.117.250 (43515)	680	655	650	900	0	0
FRA – 173.194.6 (15169)	0	0	0	380	330	320
FRA – 173.194.18 (15169)	0	0	0	370	300	230
FRA – 173.194.19 (15169)	0	0	0	260	200	165
FRA – 173.194.62 (15169)	0	0	0	325	250	275
FRA – 74.125.99 (15169)	0	0	0	190	190	135
FRA – 74.125.218 (15169)	1360	1340	1240	1260	1300	1280
FRA – 208.117.254 (43515)	670	850	0	610	1150	1380
LHR – 173.194.5 (15169)	1820	1660	0	940	1030	1170
LHR – 74.125.4 (15169)	1850	1040	1360	0	0	0
LHR – 74.125.105 (15169)	1550	1190	940	990	1060	1130
PAR – 208.117.236 (43515)	165	540	500	0	0	0
ZRH – 74.125.13 (15169)	1075	2265	1370	0	0	0

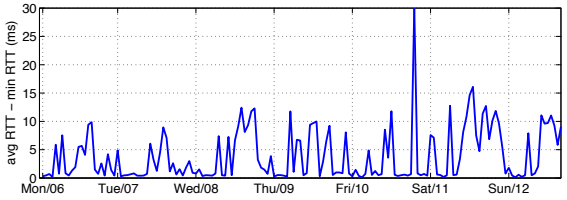
Figure 18. Daily average download throughout of YouTube flows per city and $/24$ subnetwork. The flows shifted to Frankfurt on the 8th of May are provisioned with a very low throughput. Colors reflect the QoE of the users (green = good, yellow = average, red = bad), based on the thresholds defined in Section IV.



(a) Median of min RTT per hour for all YouTube flows.



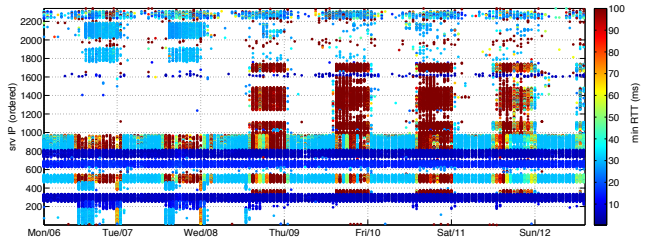
(b) Median of HTTP elaboration time per hour for all YouTube flows.



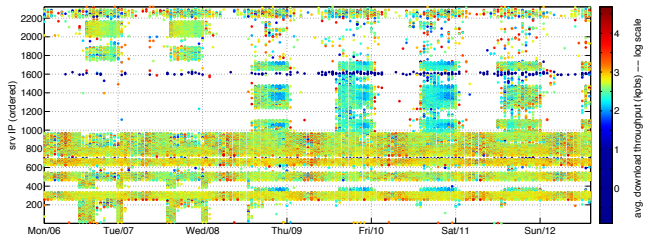
(c) Median of avg RTT - min RTT per hour for all YouTube flows.

Figure 17. 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 avg. queuing delay (difference between avg. and min. RTT) remains bounded during the anomaly, so we discard the hypothesis of path congestion.

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 Figure 17(b). To understand if these latency increases are additionally caused by path congestion, Figure 17(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



(a) Average min RTT per server IP.

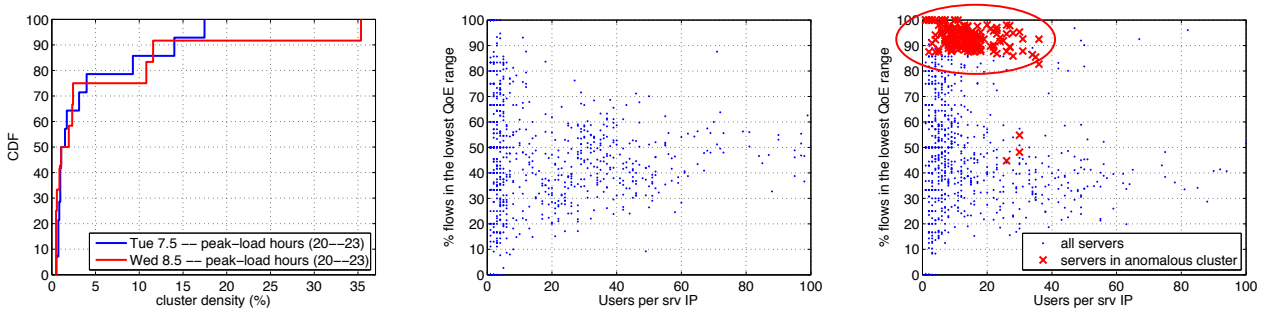


(b) Average download flow throughput per server IP.

Figure 19. 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.

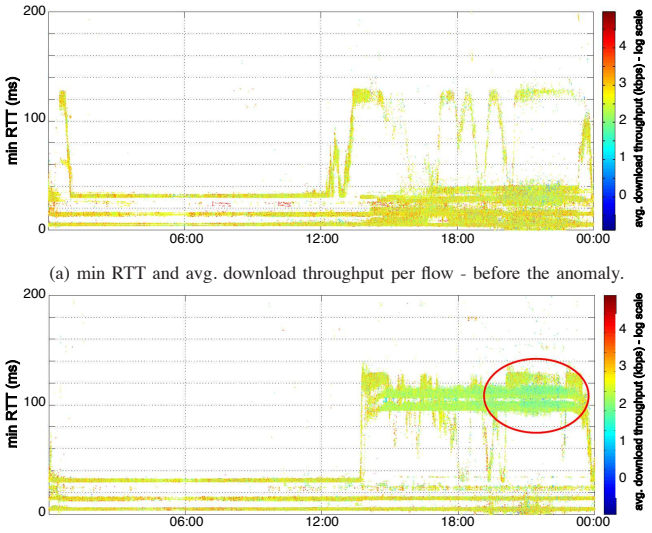
congestion. This is also confirmed by the analysis of the packet retransmissions, which do not present significant variations. Indeed, by applying the techniques we have recently presented in [21], we were not able to identify the presence of a capacity bottleneck on the downstream paths.

The last part of the diagnosis focuses on the YouTube servers. Figure 18 depicts the daily average download throughout of YouTube flows per city and per $/24$ subnetwork, using the geo-localization information described before. The color of each geo-temporal slot reflects the QoE of the users accessing the corresponding servers, based on the thresholds defined in Section IV (green = good QoE, yellow = average QoE, red = bad QoE). As expected, the shift depicted in Figure 16 from Amsterdam to Frankfurt is accompanied by a very strong degradation on the QoE of the users.



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

Figure 21. Unsupervised detection of the anomaly through clustering. There is a clear shift in the cluster density during the hours of the anomaly.



(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 20. 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.

Figure 19 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, Figure 19(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. Figure 20 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.

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

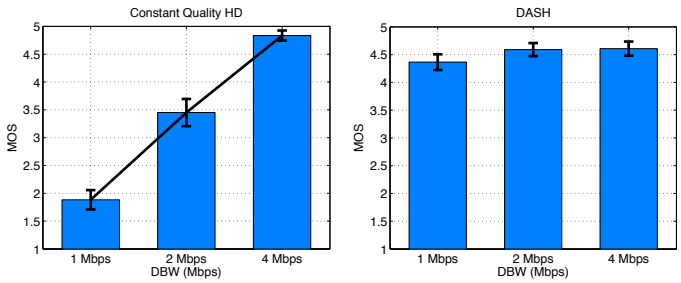
D. Unsupervised Analysis

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.

Figure 21(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 Figures 21(b) and 21(c), the newly observed cluster corresponds to a set of server IPs providing a large share of YouTube 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.

VII. And What About YouTube QoE in DASH?

So far we have focused the QoE analysis of YouTube on the fixed-quality video streaming approach followed by YouTube in the past. In this context, the video coding and video bitrate are constant during the complete streaming/watching of a video. However, the massive application of YouTube Dynamic Adaptive Streaming (DASH) in today's Internet introduces new challenges in the YouTube QoE-based monitoring from network measurements as presented in this paper. In particular, YouTube DASH uses multiple different video resolutions during a single video playback, resulting in multiple different



(a) QoE – constant HD quality. (b) QoE – DASH.

Figure 22. Overall QoE for YouTube, considering both constant HD quality and DASH. Videos are HD 720p. While DASH is able to cope with stallings by reducing the video resolution, the constant quality scenario results in very low QoE when bandwidth is not high enough.

video bitrate values. As such, the definition of the β KPI is no longer valid and needs to be updated for future studies.

To show how different could potentially be the QoE of a user watching YouTube in constant HD vs. YouTube DASH, we have conducted a subjective study where 53 participants provided their feedback in terms of experience and satisfaction with YouTube in both scenarios, while shaping down the traffic throughput between 1 Mbps and 4 Mbps. Figure 22 reports the overall quality results obtained in this subjective study. In the YouTube constant HD scenario, different HD 720p resolution videos are watched at constant quality, whereas in the DASH case, the same videos are originally requested in HD 720p, but are then dynamically adapted by YouTube itself in case of bandwidth variations. Figure 22(a) shows the results for the traditional constant quality scenario. Note that in this scenario, the β-based QoE monitoring approach works very well: 720p HD videos are usually encoded with video bitrates around 3 Mbps, which means about 4 Mbps to avoid stallings according to the results of Figure 2 (i.e., 3 Mbps $\times$ 1.25 $\approx$ 3.8 Mbps). However, it is quite impressive to see how the DASH approach results in a nearly optimal QoE for all the tested conditions (from 1 Mbps to 4 Mbps). The main difference here is that DASH changes the video quality without incurring in playback stallings, whereas the fixed quality configuration definitely results in video stallings.

As a major take away from this simple yet interesting subjective study, we have to evolve the definition of the β KPI to additionally capture the QoE of those users watching YouTube in DASH, which is becoming the default option set by the YouTube player. We are currently working on such an extended KPI.

VIII. 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[4], 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, statistical distribution-based analysis, 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 addition, we presented a structured approach for partially diagnosing CDN-related issues, which even if not complete, it shows the complexity behind the tackled problem. Finally, by conducting subjective tests on the QoE of YouTube using DASH, we showed that the QoE-based analysis of YouTube is nothing but close, and further studies are required to properly address the problem from network measurements. 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.

Although this paper has focused on the Google CDN, other CDNs are based on exactly the same principles, so we expect the analysis procedure and take aways of this work to be applicable to other CDNs. The same applies to other services, taking into account their different requirements in terms of network Quality of Service and QoE. The measurement approach followed in our work is currently being implemented within a more generic platform for automatic troubleshooting support. This platform is called mPlane: in the EU project mPlane[5] we are building a global Internet-scale measurement platform to better understand and diagnose performance degradation events in Internet-scale services such as YouTube, Facebook, Netflix, and others.

Acknowledgments

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627, "mPlane" project. The work 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. We would like to thank the anonymous reviewers for their detailed comments and suggestions, which helped us to significantly improve the quality of the paper.

References

[1] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang. Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery. In *INFOCOM, 2012 Proceedings IEEE*, 2012.

[2] A. Bär, A. Finamore, P. Casas, L. Golab, and M. Mellia. Large-scale Network Traffic Monitoring with DBStream, a System for Rolling Big Data Analysis. In *BigData, 2014 Proceedings IEEE*, 2014.

[3] P. Casas, A. D'Alconzo, P. Fiadino, A. Bär, and A. Finamore. On the Analysis of QoE-based Performance Degradation in YouTube Traffic. In *Network and Service Management (CNSM), 2014 10th International Conference on*, 2014.

[4] P. Casas, P. Fiadino, and A. Bär. IP Mining: Extracting Knowledge from the Dynamics of the Internet Addressing Space. In *Teletraffic Congress (ITC), 2013 25th International*, 2013.

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

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

[5] P. Casas, J. Mazel, and P. Owezarski. Unada: Unsupervised Network Anomaly Detection using Sub-space Outliers Ranking. In *NETWORKING, 2011 Proceedings IFIP TC 6*, Volume Part I, pp. 40–51, 2011. Springer-Verlag.

[6] P. Casas, A. Sackl, S. Egger, and R. Schatz. YouTube & Facebook Quality of Experience in Mobile Broadband Networks. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, 2012.

[7] P. Casas, M. Seufert, and R. Schatz. YOUQMON: A System for On-line Monitoring of YouTube QoE in Operational 3F Networks. *SIGMETRICS Perform. Eval. Rev.*, vol. 41(2), pp. 44–46, 2013.

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[9] A. D'Alconzo, A. Coluccia, and P. Romirer-Maierhofer. Distribution-based Anomaly Detection in 3G Mobile Networks: From Theory to Practice. *Int. J. Netw. Manag.*, vol. 20(5), pp. 245–269, 2010.

[10] M. Ester, H. peter Kriegel, J. S, and X. Xu. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ACM KDD '96. AAAI Press, 1996.

[11] P. Fiadino, A. D'Alconzo, A. Bär, A. Finamore, and P. Casas. On the Detection of Network Traffic Anomalies in Content Delivery Network Services. In *Teletraffic Congress (ITC), 2014 26th International*, 2014.

[12] A. Finamore, M. Mellia, M. Meo, M. Munafo, and D. Rossi. Experiences of Internet Traffic Monitoring with Tstat. *Network, IEEE*, vol. 25(3), pp. 8–14, 2011.

[13] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pp. 345–360, New York, NY, USA, 2011. ACM.

[14] V. Gehlen, A. Finamore, M. Mellia, and M. M. Munafò. Uncovering the Big Players of the Web. In *Proceedings of the 4th International Conference on Traffic Monitoring and Analysis*, TMA'12, pp. 15–28, 2012. Springer-Verlag.

[15] C. Huang, A. Wang, J. Li, and K. W. Ross. Measuring and Evaluating Large-scale CDNs paper Withdrawn at Microsoft's Request. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, pp. 15–29, New York, NY, USA, 2008 (Withdrawn). ACM.

[16] J. Jiang, V. Sekar, I. Stoica, and H. Zhang. Shedding Light on the Structure of Internet Video Quality Problems in the Wild. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pp. 357–368, New York, NY, USA, 2013. ACM.

[17] R. Krishnan, H. V. Madhyastha, S. Srinivasan, S. Jain, A. Krishnamurthy, T. Anderson, and J. Gao. Moving Beyond End-to-End Path Information to Optimize CDN Performance. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pp. 190–201, New York, NY, USA, 2009. ACM.

[18] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-domain Traffic. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pp. 75–86, New York, NY, USA, 2010. ACM.

[19] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies using Traffic Feature Distributions. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '05, pp. 217–228, New York, NY, USA, 2005. ACM.

[20] A. Liotta, V. Menkovski, G. Exarchakos, and A. C. Sánchez. Quality of Experience Models for Multimedia Streaming. *Int. J. Mob. Comput. Multimed. Commun.*, vol. 2(4), pp. 1–20, 2010.

[21] M. Schiavone, P. Romirer-Maierhofer, F. Ricciato, and A. Baiocchi. Towards Bottleneck Identification in Cellular Networks via Passive TCP Monitoring In *Proceedings of the 13th International Conference on Ad Hoc Networks and Wireless*, ADHOC-NOW'14, pp. 72–85, 2014. Springer International Publishing.

[22] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang. An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, pp. 151–156, New York, NY, USA, 2008. ACM.

[23] E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai Network: A Platform for High-performance Internet Applications. *SIGOPS Oper. Syst. Rev.*, vol. 44(3), pp. 2–19, 2010.

[24] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering What-if Deployment and Configuration Questions with WISE. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, pp. 99–110, New York, NY, USA, 2008. ACM.

[25] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting Video Server Selection Strategies in the YouTube CDN. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, ICDCS '11, pp. 248–257, 2011. IEEE Computer Society.

[26] M. Yu, W. Jiang, H. Li, and I. Stoica. Tradeoffs in CDN Designs for Throughput Oriented Traffic. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, pp. 145–156, New York, NY, USA, 2012. ACM.

[27] Y. Zhu, B. Helsley, J. Rexford, A. Siganporia, and S. Srinivasan. Latlong: Diagnosing Wide-Area Latency Changes for CDNs. *Network and Service Management, IEEE Transactions on*, vol. 9(3), pp. 333–345, 2012.

[28] M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of YouTube Network Traffic at a Campus Network - Measurements, Models, and Implications. *Comput. Netw.*, vol. 53(4), pp. 501–514, 2009.

Pedro Casas is Senior Researcher at the Telecommunications Research Center Vienna (FTW), Austria. He received an Electrical Engineering degree from the University of the Republic, Uruguay in 2005, and a Ph.D. degree in Computer Science from Télécom Bretagne, France in 2010. He held a Research and Teaching Assistant position at the University of the Republic between 2003 and 2012, and was at the french research lab LAAS-CNRS in Toulouse as a Postdoctoral Research Fellow between 2010 and 2011. His main research areas span the monitoring and analysis of network traffic, network security and anomaly detection, QoE modeling and automatic assessment, as well as machine-learning and data mining based approaches for Networking. Dr. Casas has published more than 60 Networking research papers (40 as main author) in major international conferences and journals, and has received 7 paper awards in the last 6 years.

Alessandro D'Alconzo received the MSc Diploma in Electrical Engineering and the Ph.D. degree from Polytechnic of Bari, Italy, in 2003 and 2007 respectively. Since 2007 he is Senior Researcher at the Telecommunications Research Center Vienna (FTW), Austria. Since 2008 he is Management Committee representative of Austria for the COST Action IC0703 "Traffic Monitoring and Analysis". He has managed and contributed to several EU projects as well as to national founded applied projects in the field of network traffic monitoring and analysis. His research interests cover various topics in the field of Telecommunication Networks, including traffic monitoring and analysis, network measurements, security and privacy, and detection and diagnosis of network traffic anomalies in Content Delivery Networks.

Pierdomenico Fiadino received his BSc and MSc degree in Computer Engineering from Sapienza University of Rome, Italy, respectively in 2008 and 2010. Since 2012, he is a PhD candidate at the Technical University of Vienna, Institute of Telecommunications. He currently works as a researcher at the Telecommunications Research Center Vienna (FTW), where he is involved in projects dealing with network analytics, anomaly detection and data mining.

Arian Bär is a PhD candidate in Computer Science in the second year at the Telecommunications Research Center Vienna (FTW). He received his Diploma degree in Computer Science from the Friedrich-Alexander Universität Erlangen-Nürnberg in 2009. His PhD topic is about the application of data base approaches to big and fast data streams common in network monitoring environments. His research interests include network monitoring and analytics, data stream warehousing, query scheduling, data mining and machine learning. He has (co)-authored about 15 publications in international journals and conferences.

Alessandro Finamore (M'09) received his Ph.D degree in Electronics and Communication Engineering in 2012 from Politecnico di Torino. In 2010 he was a visiting student at Purdue University while between 2011 and 2012 he was an intern at Telefonica Research in Barcelona. From 2012 he is a research assistant at Politecnico di Torino. His current research interests include Internet traffic measurements, CDN services, and Big Data technologies.

Tanja Zseby is a professor of communication networks at the faculty of electrical engineering and information technology at Vienna University of Technology. She received her Dipl.-Ing. degree in electrical engineering and her Dr.-Ing. degree from Technical University Berlin, Germany. Before joining Vienna University of Technology she worked as scientist and group leader at the Fraunhofer Institute for Open Communication Systems (FOKUS) in Berlin and as a visiting scientist at the University of California, San Diego (UCSD).