Who to Blame when YouTube is not Working? Detecting Anomalies in CDN-Provisioned Services

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

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

***Abstract*—Internet-scale services like YouTube are provisioned by large Content Delivery Networks (CDNs), which push content as close as possible to the end-users to improve their Quality of Experience (QoE) and to pursue their own optimization goals. Adopting space and time variant traffic delivery policies, CDNs serve users' requests from multiple servers/caches at different physical locations and different times. CDNs traffic distribution policies can have a relevant impact on the traffic routed through the Internet Service Provider (ISP), as well as unexpected negative effects on the end-user QoE. In the event of poor QoE due to faulty CDN server selection, a major problem for the ISP is to avoid being blamed by its customers. In this paper we show a real case study in which Google CDN server selection policies negatively impact the QoE of the customers of a major European ISP watching YouTube. We argue that it is extremely important for the ISP to rapidly and automatically detect such events to increase its visibility on the overall operation of the network, as well as to promptly answer possible customer complaints. We therefore present an Anomaly Detection (AD) system for detecting unexpected cache-selection changes in the traffic delivered by CDNs. The proposed algorithm improves over traditional AD approaches by analyzing the complete probability distribution of the monitored features, as well as by self-adapting its functioning to dynamic environments, providing better detection capabilities.**

Keywords*—*CDNs, YouTube, Google, QoE, Anomaly Detection, Empirical CDFs, Kullback-Leibler Divergence.

I. Introduction

Content Delivery Networks (CDNs) are a vital part of current Internet infrastructure, as major CDNs host a large share of today's Internet traffic [3]–[5]. Massively distributed server infrastructures are deployed to replicate content and make it accessible from different Internet locations. For example, Akamai operates more than 137.000 servers in more than 85 countries across nearly 1.200 networks[1], Google operates tens of data-centers and server clusters worldwide [6], and other companies such as Microsoft, Amazon, and Limelight follow similar approaches with highly distributed infrastructures. This scenario is not expected to change significantly in the next years. For instance, Google infrastructures presented a sevenfold increasing over just one year [1], while Cisco forecast that 51% of all Internet traffic will be served by CDNs by 2017 [2].

The intrinsic distributed nature of CDNs allows to better cope with the ever-increasing users' content demand. Popular applications and contents are pushed as close as possible to end-users to reduce latency and improve Quality of Experience (QoE). Load balancing policies are commonly used to limit servers load, handle internal outages, help during services 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 large CDNs allow 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[2]. Through this service, users can compare statistic 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 issues. In particular, in this paper we report an anecdotal case 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[3]. This event shows that actually Google itself might be responsible for YouTube service degradation.

This underlines the need of efficient Anomaly Detection (AD) algorithms to rapidly and effectively detect such events, both for the content provider and the ISP. There has been a considerable amount of work on AD for network traffic. We refer the reader to [8]–[10] and the references therein for a comprehensive overview on the subject. Similarly, several studies characterize CDNs architectures and focus on the optimization of their performance, servers location, and latencies [6], [7]. However, despite these efforts, to the best of our knowledge little has been done to combine these two research areas towards designing specific AD algorithms addressing the detection of unexpected cache selection events in CDNs.

This work extends the methodology we introduced in [8] to the case of AD in CDN services. Our approach considers the

[1] http://www.akamai.com/html/about/facts_figures.html

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

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

978-1-4799-0959-9/14/$31.00 ©2014 IEEE

entire distribution of different traffic features across individual CDN servers, rather than only specific moments of the random variable distributions (e.g., mean-based, percentile-based, or variance-based change detection). More in details, we rely on a continuous comparison over time of traffic features empirical distributions to identify anomalous deviations, applying an extended Kullback-Leibler similarity metric.

The contribution of this paper is twofold: firstly, we report and analyze the occurrence of a real YouTube service degradation event caused by the server selection strategies employed by Google. By explicitly showing that events in which CDN server selection policies result in poor end-user experience actually occur, we put part of the blame on Google itself when YouTube is not properly working, indirectly suggesting that initiatives such as the Google VQR should also report their own performance. Secondly, we present an approach to rapidly and automatically detect the occurrence of such anomalous events, and evaluate its functioning on the specific YouTube issue. The underlying AD algorithm is capable of dynamically adapting its definition of normal operation traffic, which results a paramount asset when considering the dynamic behavior of CDN-based traffic delivery.

The reader should note that this paper focuses exclusively on the detection of the aforementioned anomalous events, and not on their mitigation. The counteractions the ISP may take once the proposed system quickly reveals the occurrence of a CDN-based anomaly is out of the scope of our study.

II. Statistical Anomaly Detection

The goal of the AD algorithm is to detect macroscopic anomalies in the aggregate traffic served by CDNs, meaning events that involve multiple flows and/or affect multiple users at the same time. For this purpose, we resort to the temporal analysis of the entire probability distributions of certain traffic descriptors or features. In a nutshell, the proposed statistical non-parametric anomaly detection algorithm works by comparing the current probability distribution of a feature f to a set of reference distributions describing its "normal" behavior. 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 address), and end-users experienced performance (e.g., flow download throughput). Features are computed on a temporal basis, considering time bins of fixed length, referred to as time scale. The following sections describe the algorithm.

A. Overview of the Algorithm

Given a traffic feature f, we define $f_i^\tau(t)$ the generic counter observed at the t-th time bin of length τ. For instance, if f represents the number of flows served by each server IP address every 5 minutes, $i \in \{1, \ldots, n(t)\}$ reflects the i-th server IP address contacted, while $f_i^\tau(t)$ counts the number of flows handled by that IP address over the t-th 5 minute time bin. The length of τ defines the timescale of the data aggregation, which in turn defines the timescale of the observable anomalous events. Given a certain time scale τ, the set of counters $\mathcal{F}^\tau(t) = \{f_i^\tau(t)\}$ can be used to derive the empirical distribution of the feature f, denoted by $X^\tau(t)$ [4]. Considering the example reported above, each 5 minutes we obtain per-IP statistics that we can use to compute the overall number of flows served, as $N(t) = \sum_i f_i^\tau(t)$. Notice that by properly grouping the same input data we can obtain statistics at different "views" on the network data (e.g., /24 subnets, Autonomous System Numbers, etc.). As the following analysis can be done independently of the specific selected time scale, we omit the superscript τ from now on.

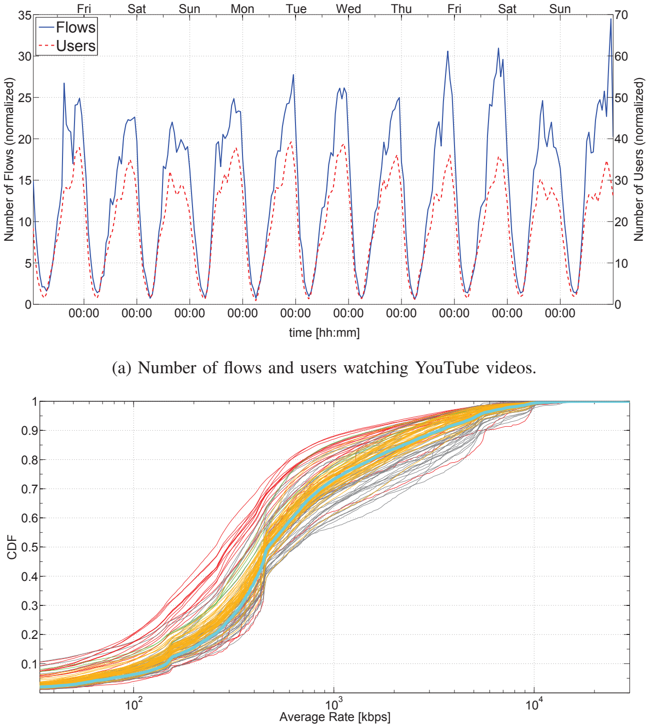
The anomaly detection algorithm consists in computing the degree of similarity between current distribution at time t, and a set of reference distributions computed from past measurements at times $t_j < t$. To construct this reference set, we introduce the notion of *observation window* $\mathcal{W}(t)$, which is simply a sliding window containing past time bins: $\mathcal{W}(t) = \{t_j : a(t) \leq t_j \leq b(t)\}$, where $a(t)$ and $b(t)$ are the oldest and the most recent time bins that can be considered to test the distribution $X(t)$ at current time t. The reference time bins set is denoted as $\mathcal{I}(t) \subseteq \mathcal{W}(t)$, and corresponds to the set of time bins selected from $\mathcal{W}(t)$ by running the *reference set identification algorithm* briefly described in section II-C. This algorithm identifies the set of past time bins with the most similar anomaly-free distributions to the current one. Given two distributions $X(t_i)$ and $X(t_j)$, of the same feature and timescale, at times t_i and t_j, we define $L(t_i, t_j)$ as a divergence metric accounting for the degree of similarity between the two of them. The choice of divergence metric is discussed next. The comparison between the current distribution $X(t)$ and the associated distributions reference set $\{X(t_j), t_j \in \mathcal{I}(t)\}$ involves the computation of two compound metrics based on the divergence $L(\cdot, \cdot)$. The first one, called *internal dispersion* and denoted by $\Phi_\alpha(t)$, is a synthetic indicator derived from the set of divergences computed between all the pairs of distributions in the reference set. Formally, $\{L(t_i, t_j), t_i, t_j \in \mathcal{I}(t), t_i \neq t_j\} \to \Phi_\alpha(t)$. We chose $\Phi_\alpha(t)$ to be the α-percentile of this set of divergence measures. The parameter α must be tuned to adjust the sensitivity of the detection algorithm: it defines the maximum distribution deviation that can be accounted to normal statistical fluctuations, therefore an acceptance region for the AD test. Similarly, we define the *external dispersion* $\Gamma(t)$ as a synthetic indicator extracted from the set of divergences between the current distribution $X(t)$ and those in the reference set. Formally, $\{L(t_i, t),\ t_i \in \mathcal{I}(t)\} \to \Gamma(t)$. We chose $\Gamma(t)$ as the mean.

The detection scheme is based on the comparison between the internal and external metrics. If $\Gamma(t) \leq \Phi_\alpha(t)$ then the observation $X(t)$ is marked as normal. In this case, the boundaries of the observation window are updated by one time bin shift. Conversely, the condition $\Gamma(t) > \Phi_\alpha(t)$ triggers an alarm, and $X(t)$ is marked as abnormal. The corresponding time bin t is then included in the set of anomalous time bins $\mathcal{M}(t)$, and is excluded from all future reference sets. In this case only the upper bound of the observation window is shifted, i.e. $a(t+1) = a(t)$ and $b(t+1) = b(t) + 1$. Such update rule is meant to prevent the reference set from shrinking in case of persistent anomalies. In fact, only the time bins in $\mathcal{W}(t) \setminus \mathcal{M}(t)$ are considered for the reference set.

B. Divergence Metric for Anomaly Detection

A possible distance metric between two distributions is the *Kullback-Leibler* (KL) divergence. Let p and q be two dis-

[4] Only non-zero counters are actually considered when deriving the empirical distributions.



(a) Number of flows and users watching YouTube videos.

(b) Output of the reference set identification procedure.

Figure 1. (a) Total number of flows related to download average rate, and number of users generating the traffic. (b) Output of the reference set identification algorithm.

crete probability distributions defined over a common discrete probability space Ω. The KL divergence is defined as [11]:

$$D(p||q) = \mathrm{E}\left[\log\left(\frac{p(\omega)}{q(\omega)}\right)\right] = \sum_{\omega\in\Omega} p(\omega)\log\left(\frac{p(\omega)}{q(\omega)}\right) \qquad (1)$$

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 if $\leftrightarrow p = q$, and for each $\omega \in \Omega$ it weights the discrepancies between p and q by $p(\omega)$. The KL divergence has several optimality proprieties that make it ideal for representing the difference between distributions [11]. However, it cannot be actually considered as a distance metric, since it is not symmetric and does not satisfy the triangular inequality. In particular, the lack of symmetry can be inconvenient in certain scenarios, particularly in the presence of events that take very low probability values in only one of the two tested distributions. Therefore, we adopted a more elaborated divergence metric, symmetric by construction:

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

where $D(\cdot||\cdot)$ is defined according to eq. (1), and H_p and H_q are the entropy of p and q respectively. The properties of this metric are extensively discussed in [8].

C. Identification of the Reference Set

The design of the algorithm considers the identification of a set of distributions, which is used as the normality reference for the detection step. The identification of a suitable reference assumes a paramount relevance in the context of CDNs' traffic AD, due to the highly dynamic way CDNs host and serve the contents. Most of the AD work considers training once-and-for-ever and tests the current sample against the most recent ones. In the context of CDN AD, a reference based only on the most recent samples would not be able to take into account the steep variation in the total traffic counters in the morning and in the late evening, resulting in a series of false alarms. From the exploration of the real traffic traces we found that the traffic served by the analyzed CDNs share some common *structural characteristics* which must be considered for the choice of the observation window and reference set. For example the traffic is non-stationary due to time-of-day variations, with steep variations occurring at certain specific hours like peak-utilization time, and with very strong 24-hours seasonality. We remark that such variations do not only apply to the flow counts and active server IPs, but also to the distribution of many other features such as volume, RTT to the servers, download throughput, etc.

The heuristic used for the construction of the reference set follows a progressive refinement approach, where the mentioned structural characteristics are used at each step for reducing the set of candidate references in the observation window $\mathcal{W}(t)$. At each step, the set of candidate references is incrementally reduced by filtering the elements according to three different criteria. Given a new sample at time t of size $N(t)$, in the first step the algorithm picks the subset $\mathcal{I}_0(t)$ of past time bins with samples of similar size, formally $\mathcal{I}_0(t) = \{j | N(t) - s \leq N(j) < N(t) + s\}$, where s is a slack factor. Such size-based criterion avoids comparing distributions with very different statistical significance, as the sample size can vary across two orders of magnitude during the 24 hours (see for example figure 1(a)). In a second refinement step, the subset of elements in $\mathcal{I}_0(t)$ with the smallest divergence from current observation are picked. In this way, samples related to different times of the day and/or type of day (working day vs. weekends/festivities) are filtered out. The residual set $\mathcal{I}_1(t)$ might still contain residual heterogeneous samples. To eliminate these samples, in the third step we resort to an heuristic in which we apply a graph-based clustering procedure to identify the dominant subset with the lowest inter-samples divergence: samples are mapped to nodes, with edges weighted proportionally to the KL divergence among them. The algorithm divides the nodes in two clusters so as to minimize the intra-cluster divergence, and finally the larger cluster is picked as the final reference set $\mathcal{I}(t)$.

The overall procedure is designed to minimize the inter-samples divergence within the reference set, so as to preserve good sensitivity of the detection process. We stress the fact that past observations (distributions), which were previously marked as "anomalous" by the detector, are excluded from the reference identification procedure. In other words, only samples marked as "normal" are taken as candidates. This introduces a feedback loop, as the output of the detector for past samples impacts the identification of the reference set, and therefore influences the future decisions.

Our experience shows that the proposed heuristic copes well with the time variability of both the distribution shape and the sample size. It does so by embedding the intrinsic pseudo-cyclical structure of the real traffic process into the reference set, resulting in a minimum set of past observations with the lowest divergence with respect to the current sample. In a nutshell, it leverages pseudo-seasonality to compensate for non-stationarity. As an example, figure 1(b) shows the typical output of the reference identification algorithm. In this specific example, we consider the distribution of the average download rate across the users watching YouTube videos during 11 consecutive days (see Sec. III for the details on this dataset). Figure 1(a) explains the ideas behind the first step of the reference set identification procedure, where distributions are selected based on the number of samples – flows in this case – used to derive them (absolute values are normalized for privacy reasons).

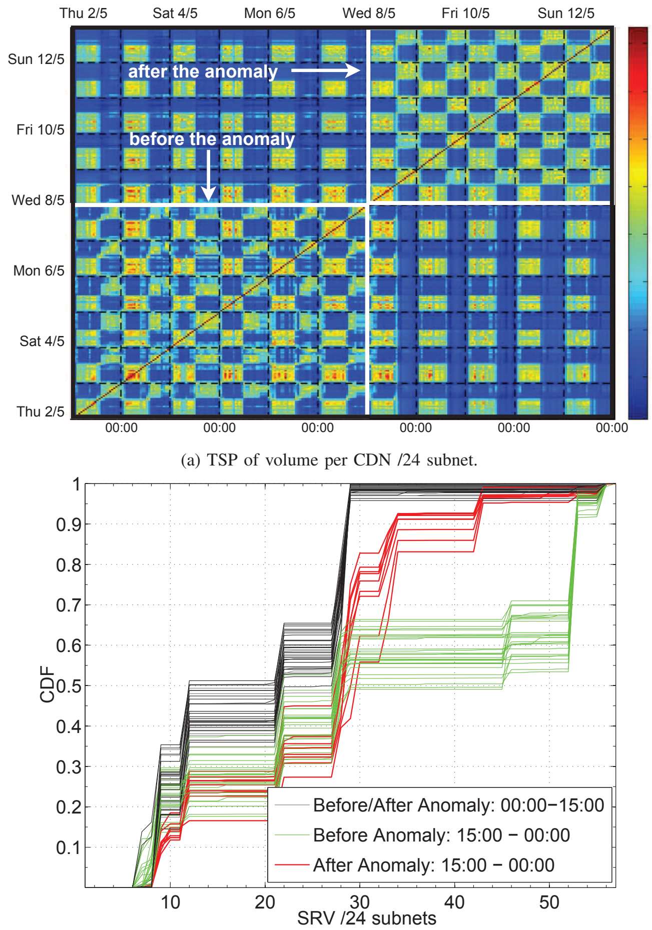
Figure 1(b) depicts the output of the reference set identification algorithm. The cyan CDF represents the sample under test. The gray CDFs correspond to those samples in the observation window which are discarded by the identification procedure. The red CDFs are the samples in the observation window which are discarded for being previously marked as anomalous. Finally, the orange CDFs are those selected as reference. Note that out of all the possible candidate distributions, the algorithm selects the ones with lowest divergence to the distribution under test, i.e., the orange CDFs. We remark that the proposed scheme is robust to irregularities in the pseudo-cycles – as introduced for example by non-weekends festivities, or solar/legal time shifts – since it does not rely on any external label information (e.g. calendar day or absolute time). For further details on the reference set identification, the interested reader is referred to [8].

III. User Experience Anomalies in YouTube

The main point we make in this paper is that CDN cache selection policies may have a strong impact on the service quality as experienced by the end users. This is not only a main issue for the end-users, but also for the ISP providing the Internet access to the contents, as customers will in most cases directly blame the ISP for the bad QoE, even if the origin of the problems is located outside its boundaries.

This section reports a real case in which an unexpected cache selection and load balancing policy employed by Google CDN results in an important drop on the average download throughput for the end-users watching YouTube videos. The ISP holding the vantage point used in this study confirmed that the effect was negatively perceived by its customers. As the issue was caused by an unexpected caches selection done by Google, the ISP internal Root Cause Analysis systems did not identify any problems inside its boundaries. As reported by the ISP operations team, the anomaly occurs on Wednesday the 8th of May.

The analyzed dataset corresponds to one month of HTTP video streaming flows collected at the fixed-line network of a major European ISP, from April the 15th till May the 14th, 2013. The monitored link aggregates about 30.000 residential customers accessing to the Internet either using ADSL or Fiber-To-The-Home (FTTH) technologies. Flows are captured using the Tstat passive monitoring system [12]. Using Tstat filtering and classification modules, we only keep those flows carrying YouTube videos. These flows are finally imported and analyzed with the data stream warehouse DBStream [14].



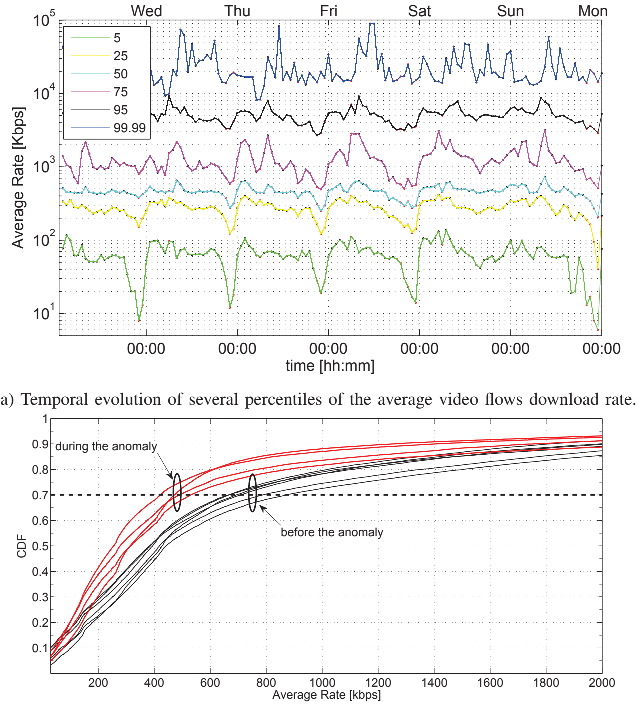
(a) TSP of volume per CDN /24 subnet.

(b) Volume per CDN /24 subnet.

Figure 2. 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, between 15:00 and 00:00.

To get some insights on the aforementioned cache selection-based anomalies, we begin by investigating the temporal evolution of the distributions of volume served by the different IPs in the dataset, aggregated in /24 subnetworks for practical reasons. By comparing the distributions referring to different time intervals through the modified K-L divergence (2), we get a direct insight on how the traffic load balancing is performed among the /24 subnetworks belonging to the Google CDN. 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 [8], 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 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. The blue palette represents low similarity values, while reddish colors correspond to high similarity values. By construction, the TSP is symmetric around the 45° diagonal.

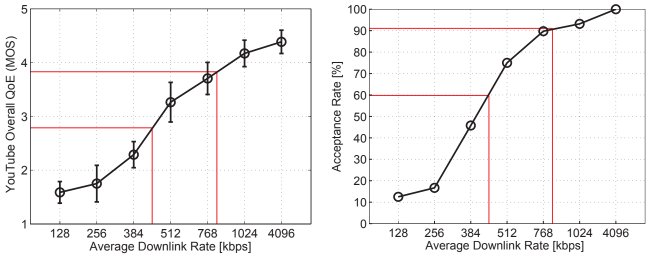
Figure 2(a) shows the TSP of the video volume served by the different IPs in the dataset, aggregated in /24 subnetworks, and using a time-scale of 1 hour. Note the regular "tile-wise"



(a) Temporal evolution of several percentiles of the average video flows download rate.

(b) CDFs before and during the anomaly.

Figure 3. Distribution of the video flows average download rate across the users: (a) trend over time for several percentiles, (b) CDFs at peak hours (21:00-23:00), before and during the reported anomaly.



(a) YouTube overall QoE vs. downlink rate.

(b) YouTube acceptability vs. downlink rate.

Figure 4. 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 [13].

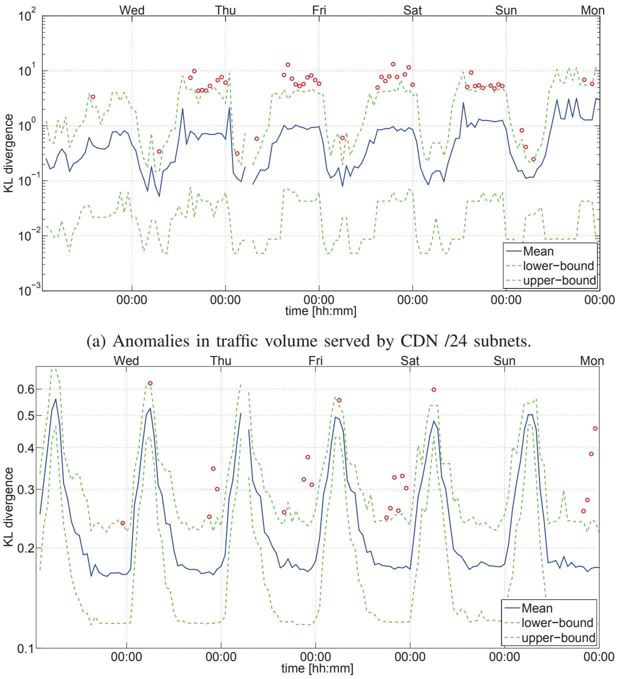
texture within a period of 24 hours, due to the daily cycle. This pattern repeats almost identical for a few days, forming multi-days macro-blocks around the main diagonal. Besides the basic tile-texture, the analysis of the entire observation period reveals the presence of a more complex temporal strategy in the (re)usage of the IP address space. Specifically, there are two subnet sets periodically re-used in the first and second half of the day. In particular, the TSP clearly reveals that a different subnet set is used during the second half of the day, from the 8th of May on. This reveals a different cache selection policy in place during the anomaly reported by the ISP operations team on Wednesday the 8th. The change is also visible in the CDFs of the per subnet volume depicted in figure 2(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.

This detected change in the cache selection policy employed by Google does not justify by itself the negative effect on the QoE of the ISP customers. To further investigate this issue, we analyze the distributions of the average video flows download rate. Figure 3(a) depicts the temporal trend of several percentiles of the average video flows download rate per user, starting one day before the anomaly occurs and covering five consecutive days after it. The lowest percentiles (i.e., 5% and 25%) show a constant drop on the average download flow rate during peak hours (between 21:00 and 23:00), even before the anomaly actually occurs. However, starting on Wednesday, even the 50% and 75% percentiles present an important drop at peak hours, which justifies the bad experience of the customers. Figure 3(b) analyzes the distribution of the average video flows download rate, in the hours before and during the anomaly. Interestingly, the only distributions exhibiting a marked change before and during the anomaly are those corresponding to the peak hours (21:00-23:00), which are those reported in figure 3(b). Indeed, if we focus for example on the 70% percentile, we observe a drastic reduction on the video flows download rate, going from about 780 kbps to 470 kbps. Even if this reduction might not look significant a priori, we know from previous QoE studies in YouTube [13], that it is sufficient to drop the perceived quality below the level of acceptance.

Figure 4 permits to explain the customers' complaints. The figure reports the overall QoE and the acceptance rate as declared by users watching YouTube videos during a field trial test conducted and reported in [13], both as a function of the average downlink 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. Both curves correspond to a best-case scenario, in which only 360p videos were watched by the users. In the anomalous situation evaluated in this paper, not only 360p videos were consumed by the customers, but most probably videos with higher resolutions (e.g., 1080p HD), and thus we expect that the impact on the user experience were even more severe than what we report in here.

Figure 4(a) shows the overall QoE as a function of the average downlink rate, using a 5-points MOS scale, where 1 corresponds to very bad QoE and 5 to optimal (note: in the practice, the dynamic range of QoE values varies between 1.5 and 4.5 MOS). The figure clearly shows that the overall QoE drops from a MOS score close to 4 at 780 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. Figure 4(b) additionally shows how the acceptance rate (i.e., the proportion of customers accepting to use the YouTube service at the corresponding downlink rate value) drops from about 90% in normal conditions to nearly 60% during the anomaly, providing more evidence on the impact of such downlink rate drop on the customers.

To conclude the analysis, we report in figure 5 the output of the proposed AD system, which automatically detects the



(a) Anomalies in traffic volume served by CDN /24 subnets.

(b) Anomalies in the video flows average download rate across the YouTube users.

Figure 5. Detection of anomalies in YouTube traffic. Alarms and acceptance region for the distribution of (a) volume and (b) video flows average download rate. The red markers correspond to the flagged anomalies.

described issue as soon as it occurs. Figure 5(a) considers the per /24 subnet served volume as the monitored feature. It shows how $\Phi_\alpha(t)$ (with $\alpha = 95$-th percentile) adapts over time to follow the natural traffic daily changes. The red markers indicate when the condition $\Gamma(t) < \Phi_\alpha(t)$ is violated, triggering an anomaly. From Wednesday the 8th of May onward the algorithm systematically rises alarms from 15:00 to 00:00, which correspond to the discussed change in the cache selection policy. Figure 5(b) reports the same information for the average video flows download rate. In this case, the AD system detects some anomalies only between peak hours (21:00-23:00) from the 8th onward, coherently with the observations drawn from figure 3. Interestingly, it can be noticed that even during peak hours, the anomalies are not detected on Saturday the 11th, whereas they are back on Sunday. This behavior is easily explained by the lower traffic served during the peak hours on Saturday, as shown in figure 1(a). Indeed, the percentiles depicted in figure 3(a) do not reveal a clear deviation on Saturday average download rates.

Comparing the changes on the volume distribution against those on the video flows download rate distribution, we observe that the cache selection policy used by Google resulted in QoE degradation only during the peak hours on the high load days. This suggests that the servers of the selected caches were not correctly dimensioned to handle traffic load peaks.

IV. Discussion and Concluding Remarks

In this paper, we have shown that the caching selection policies employed by a major CDN such as Google might have an important impact on the end-customers QoE. Our study is based on the analysis of traffic from a large dataset collected at a major ISP in Europe. The reported poor QoE by the customers suggests that it is important for the ISP to rapidly and automatically detect the occurrence of abrupt changes in caches selection policies affecting the service performance of their customers, and therefore we presented a network AD system for CDNs' traffic. By applying this algorithm to the traffic datasets, we were able to automatically identify those changes in the Google CDN cache selection policy affecting the end-user perceived quality. 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 to be blamed for poor end-user experience in Internet-scale services like YouTube.

The results presented in this paper are still in an early stage, as we are only reporting and analyzing the occurrence of a single event, using data from a single vantage point. In this paper we have not fully evaluated the limitations of our AD system to cope with the high complexity of the considered scenarios. The integration of more rich sources of data for better diagnosis of the detected anomalies, as well as a deep study of the performance of the AD algorithm are part of our ongoing work.

Acknowledgments

This work has been performed in the framework of the EU-IP project mPlane, funded by the EC under grant 318627. The Competence Center FTW Forschungszentrum Telekommunikation Wien GmbH is funded within the program COMET (Competence Centers for Excellent Technologies) by BMVIT, BMWA, and the City of Vienna. The COMET program is managed by the FFG.

References

[1] M. Calder et al. "Mapping the Expansion of Google's Serving Infrastructure" in *IMC*, 2013.

[2] Cisco Systems, "Cisco Visual Networking Index: Forecast and Methodology, 2012-2017", *white paper*, 2013.

[3] A. Gerber et al., "Traffic Types and Growth in Backbone Networks", in *OFC/NFOEC*, 2011.

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

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

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

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

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

[9] P. Casas et al., "Optimal Volume Anomaly Detection and Isolation in Large-Scale IP Networks using Coarse-Grained Measurements", in *COMNET* 54(11), 2010.

[10] M. Thottan et al., "Anomaly Detection Approaches for Communication Networks", in *Algorithms for Next Generation Networks*, Springer, 2010.

[11] J.A.T. Thomas et al., "Elements of Information Theory", Wiley & Sons, Ed., 1991.

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

[13] P. Casas et al., "YouTube & Facebook Quality of Experience in Mobile Broadband Networks", in *QoEMC*, 2012.

[14] A. Bär et al., "DBStream: an Online Aggregation, Filtering and Processing System for Network Traffic Monitoring", in *TRAC*, 2014.