Proceedings of the 2014 26th International Teletraffic Congress (ITC)

On the Detection of Network Traffic Anomalies in Content Delivery Network Services

Pierdomenico Fiadino†, Alessandro D'Alconzo†, Arian Bär†, Alessandro Finamore*, Pedro Casas†
† FTW - Telecommunications Research Center Vienna; * Politecnico di Torino
{surname}@ftw.at; finamore@tlc.polito.it

***Abstract*—Today's Internet traffic is largely dominated by major content providers and highly distributed Content Delivery Networks (CDNs). Internet-scale applications like Facebook and YouTube are served by large CDNs like Akamai and Google CDN, which push content as close to end-users as possible to improve the overall performance of the applications, minimize the effects of peering point congestion and enhance the user experience. The load is balanced among multiple servers or caches according to non-disclosed CDN internal policies. As such, adopting space and time variant policies, users' requests are served from different physical locations at different time. Cache selection and load balancing policies can have a relevant impact on the traffic routed by the underlying transport network, as well as on the end-user experience. In this paper, we analyze the provisioning of two major Internet applications, namely Facebook and YouTube, in two datasets collected at major European Internet Service Providers (ISPs). First, we show how the cache selection performed by Akamai might result in higher transport costs for the ISP. Second, we present evidence on large-scale outages occurring in the Facebook traffic distribution. Finally, we characterize the variation of YouTube cache selection strategies and their impact on the users' quality of experience. We argue that it is important for the ISP to rapidly and automatically detect such events. Therefore, we present an Anomaly Detection (AD) system for detecting unexpected cache-selection events and 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, providing higher visibility and better detection capabilities.**

***Keywords*—Anomaly Detection, Empirical CDFs, Kullback-Leibler Divergence, CDNs, YouTube, Facebook, Akamai, Google**

I. Introduction

Content Delivery Networks (CDNs) are a vital part of current Internet infrastructure. A large share of today's Internet traffic is hosted by major CDNs [2], [3], and Cisco forecasts that 51% of all Internet traffic will be served by CDNs by 2017 [1]. 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 [4], and other companies such as Microsoft, Amazon, and Limelight follow similar approaches with highly distributed infrastructures.

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 the 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. If on the one hand the highly distributed server deployment and adaptive behavior of large CDNs allows them to achieve high availability and performance, on the other hand they 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 the different ISP network paths. As a result, the traffic engineering policies of the ISP might be overruled by the CDN caching selection policies, potentially resulting in extra transport costs for the former. Finally, there might be cases in which the strategies in place result non optimal for end-users' QoE.

By tracking both the Facebook traffic served by Akamai, as well as the YouTube traffic served by Google, we show that these types of anomalous events actually occur in large CDNs. The datasets we analyze come from both a mobile and a fixed-line major European ISPs, highly enriching the conducted study. Based on our analysis, we argue that it is important for the ISP to rapidly and automatically detect the occurrence of such events. Even though multiple network Anomaly Detection (AD) approaches have been proposed in the past, to the best of our knowledge none of them has specifically addressed the complex case of CDN traffic. Therefore, we present a novel network AD approach, specialized on CDN traffic. The approach builds on our previous work on network AD [7], but is tailored towards CDN AD.

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.

The remainder of the paper is organized as follows: Sec. II provides a summary on the characterization and analysis of CDNs, and a brief overview of the network AD domain. In Sec. III we describe the proposed AD approach, tailored to detect the aforementioned unexpected events. Sec. IV presents a study of the Facebook traffic delivery through Akamai, where we identify unexpected load balancing events which result in additional transport costs for the monitored ISP, as well as service outages. In Sec. V we analyze an anomaly occurring in the YouTube CDN, which directly impacts the QoE of the end-users watching YouTube videos. Finally, Sec. VI concludes this work.

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

978-0-9836283-9-2

II. Related Work

The study of the Internet traffic and applications delivered by the top CDNs has gained important momentum in the last few years [3]–[5]. For example, [3] shows that most of today's inter-domain traffic flows directly between large content providers, CDNs, and the end-users, and that more than 30% of the inter-domain traffic volume is delivered by a small number of large CDNs and content providers. Several studies have focused on CDN architectures and CDN performance, analyzing features such as CDN size, servers' location, and latencies to content among others [4], [5]. In particular, [4] focuses on user-content latency analysis at the Google CDN, and [5] provides a comprehensive study of the highly distributed Akamai CDN architecture. Despite the large literature, none of these studies has considered the problem of detecting and analyzing anomalies in such CDN scenarios. A first step in this direction was recently taken by us in [17], where we studied the problem of anomalies in the YouTube service. Current paper extends this work by deeper studying the detected problems in YouTube, as well as targeting the Facebook case study, and specially using analysis techniques based on visual inspection and statistical data processing.

There has been a considerable amount of papers on Anomaly Detection (AD) in network traffic. We refer the reader to [7], [8] and the references therein for a comprehensive overview on the subject. However, to the best of our knowledge, none of them has specifically addressed the detection of anomalies in CDNs' delivered traffic, induced by unexpected cache selection events. Traditional approaches for network AD consider individual and independent time series analysis, processing different traffic descriptors or features with classical forecasting and outliers analysis methods. Our approach is intrinsically more powerful, as it considers the 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). A few works consider the temporal distribution of traffic volume-derived features, but these fail to detect events that do not cause appreciable changes in the total traffic volume. Our work is close to the approaches proposed in [9]–[11], where windowed temporal distributions are computed and compared through the standard Kullback-Leibler divergence in the quest for anomalous deviations. Finally, the AD algorithm builds on our previous work on network AD [7], but specifically targeting the CDN case.

III. 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. General Overview of the Algorithm

Given a certain traffic feature f (e.g., flow counts), we define $c_i^\tau(t)$ as a generic counter associated to f. The i-th counter can be associated to the client IP, to the server IP/network of a CDN, or finally to the i-th (quantized) throughput value. The symbol τ indicates the size of the time bin, and t is the time index. For example, $c_i^\tau(t)$ could be the number of flows served from IP i at time bin t of length τ minutes. 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 non-zero counters $\mathcal{C}^\tau(t) = \{c_i^\tau(t),\, i = 1, 2, \ldots, N^\tau(t)\}$ can be used to derive the empirical distribution of the feature f, denoted by $X^\tau(t)$, where the cardinality $N^\tau(t)$ could be for example the number of IPs serving traffic in the t-th time bin. 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 references 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 evaluate 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 III-D. 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, $\{L(t_i, t_j), t_i, t_j \in \mathcal{I}(t), t_i \neq t_j\} \rightarrow \Phi_\alpha(t)$. 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, $\{L(t_i, t),\ \ t_i \in \mathcal{I}(t)\} \rightarrow \Gamma(t)$. 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 discrete probability distributions defined over a common discrete probability space Ω. The KL divergence is defined as [13]:

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

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 $\leftrightarrow p = q$, 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 [13]. However, it can not 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:

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

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 [7].

C. Deriving Empirical Distributions

The feature distributions p, q in eq. (2) are unknown, hence they must be empirically obtained from the data samples. Some issues may arise in the estimation of the discrete probability distributions. Indeed, when the traffic distribution is computed for example from per IP counters, an obvious problem is the cardinality of the probability space Ω. A simple solution in this case is to consider per sub-network counters. Instead, when the considered traffic feature is the distribution of the throughput or the RTT across the users, then the empirical distributions found in real datasets are often heavy-tailed and span over ranges of a few orders of magnitude. In many cases, the sample size $N(t)$ is smaller than the range of spanned values. The standard approach in this case is to apply binning, i.e. to quantize the spanning range of the variable into a reduced number of bins, and to take the frequency of samples in each bin as the estimate of the distribution. The choice of the binning is critical because it affects the accuracy of the estimate, and ultimately the sensitivity of the detector. When this is the case, we adopt a non-uniform lin-log binning where the lower range is binned linearly and the upper one logarithmically, and the edges are automatically adapted so as to obtain a fixed number of bins. In some other cases we use our domain knowledge in defining meaningful bin edges. For example, in the case of the video download rate, we define bin edges which correspond to changes in the perceived user experience [16].

D. 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 (Akamai and Google CDN) share some common *structural characteristics* which must be considered for the choice of the observation window and reference set. For example (see Fig. 3 on Akamai traffic), 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, minimum RTT to the servers, download throughput, etc.

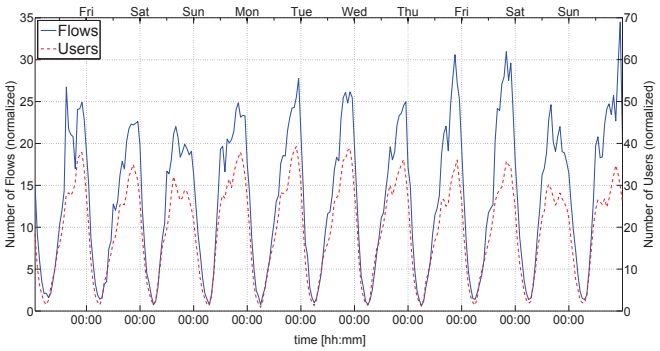
The heuristic used for the construction of the reference set follows a progressive refinement approach, where the mentioned structural characteristics are used at each step for reducing the set of candidate references in the observation window $\mathcal{W}(t)$. At each step, the set of candidate references 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 $\mathcal{I}_0(t) = \{j | N(t) - s \leq N(j) < N(t) + s\}$. 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 Fig. 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 time of 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 weights, 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

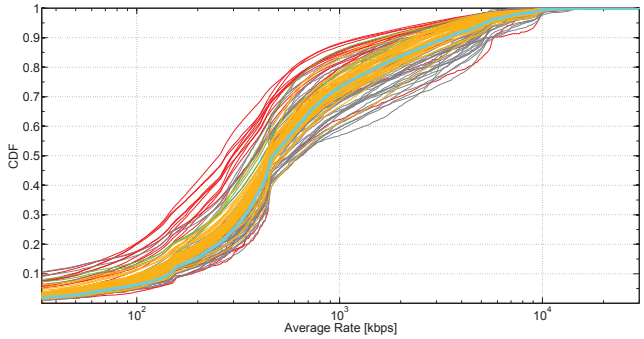
good sensitivity of the detection process. We stress the fact that past observations (samples) which were previously marked as "anomalous" by the detector are excluded from the reference identification procedure; in other words, only samples marked as "normal" are taken as candidates. This introduces 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, Fig. 1 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. V for the details on this dataset). Fig. 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 issues). As we shall see in Sec. V, the observed variations on the number of flows influences the overall behavior of the detection algorithm.

Fig. 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 current one, 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 [7].



(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.

IV. Anomalies in Akamai and Facebook

In our first case study, we analyze the well known and highly popular social network Facebook. Due to its high number of users and volume of the served traffic, Facebook content is delivered through a sophisticated and highly distributed content delivery infrastructure. The big majority of the Facebook content is hosted by the Akamai CDN. Parts of the content are hosted under Facebooks own AS, split between its headquarters in the USA and Ireland. Finally, an important share of the content is served by the ISPs, which maintain large transparent caches, and may additionally host Akamai servers inside their premises.

The dataset corresponds to one month of HTTP flow traces collected at the 3G mobile network of a major European ISP by mid 2013. Flows are captured at the Gn interface and imported and analyzed with the stream data warehouse DB-Stream [18]. Facebook flows are filtered using the HTTPTag traffic classification tool [12]. To preserve user privacy, any user related data (e.g., IMSI, MSISDN) are removed on-the-fly, and payload content beyond HTTP headers is discarded. Using the MaxMIND ASes databases[2], the ASes serving the corresponding flows are included in the dataset. In the following analysis, dates are not disclosed and flow/volume counts are normalized to preserve business privacy.

Let us first show with a simple example the intrinsic multi-caches and daily load balancing policies employed in the delivery of Facebook traffic flows. Fig. 2 shows the per-hour distribution (CCDF) of the RTT of the flows carrying Facebook content for a complete day. For each Facebook flow, the RTT is passively computed as the delay between the SYN and the SYNACK packets during the TCP 3-way handshake. Given that the probe is at the Gn interface of a 3G mobile network, the user-side part of the RTT is excluded. For further details about the metering methodology we refer the reader to [15]. The Fig. reveals the typical daily patterns of the RTT distributions. The occurrence of "bumps" or knees in the distribution indicates the presence of different caches, located at different propagation distances from the vantage point. In addition, there is a clear change on the selected servers providing the content during the first and the second half of the day, revealing the existence of a time-of-day based load balancing policy.

[2]MaxMIND databases, http://www.maxmind.com.

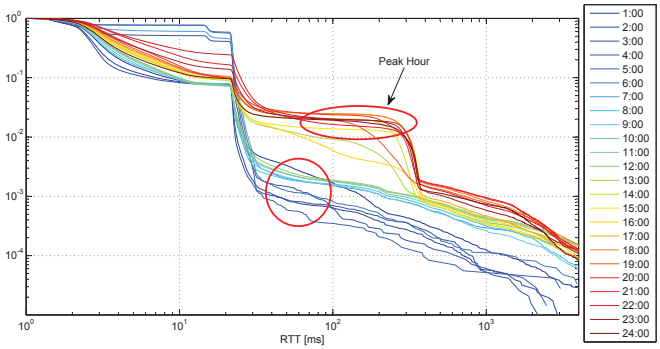


Figure 2. Daily RTT CCDFs for Facebook flows. There is a clear shift on the selected servers between the first and the second half of the day.

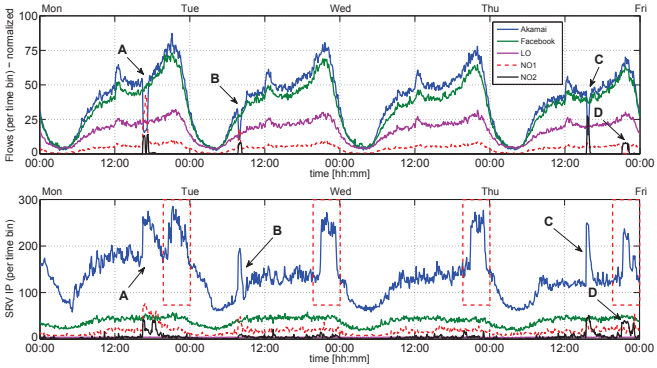


Figure 3. Flow counts (up) and server IPs (down) per AS, 5-min aggregation.

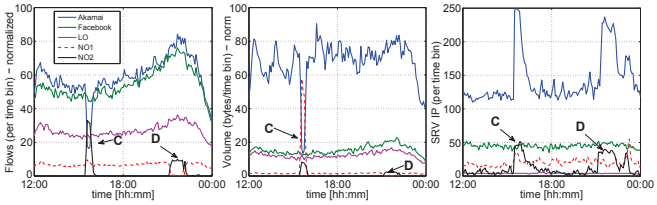


Figure 4. Flow counts, volume and server IPs per AS, for 12 hours.

Let us move now to the core of the analysis. To show some of the aforementioned unexpected traffic changes caused by the selection of caches serving Facebook, Fig. 3 depicts the 4-days evolution (day 21 - day 24) of the number of flows and the corresponding number of unique server IPs delivering Facebook content, aggregated in 5-min time bins and split by hosting organization/AS. We include the top-4 organizations in terms of delivered volume, which correspond to Akamai, Facebook AS, the Local Operator (LO), and the most important Neighbor Operator (NO1). The plot also includes another Neighboring Operator we refer to as NO2, which plays a key role in this analysis. The flow share across the 5 organizations remains practically constant during the day. There is a clear daily pattern in the number of active IPs, and it is worth noting how Akamai systematically doubles the number of deployed servers during the peak hours (21:00-23:00), flagged by the dotted rectangles. As expected, Akamai and Facebook AS serve the largest share of Facebook flows. Akamai employs many more servers, and as shown in Fig. 4, it hosts the largest flows corresponding to the static Facebook contents, showing the role breakdown through the different organizations.

Fig. 3 additionally shows the occurrence of four anomalies, identified as A, B, C and D, which break the normal traffic pattern. We clarify to the reader that these events are assessed as "unexpected" or anomalous with respect to the behavior observed in our traces, i.e., from the perspective of the ISP hosting the vantage point. In this study we do not have enough data (e.g., from multiple vantage points) to find the root causes of such behaviors, which might be the result of more complex and planned activities by the involved ASes. Anomalies A and B have similar characteristics: even if the number of IPs steeply increases, the number of flows and traffic volume served by Akamai abruptly decreases. The number of flows served from NO1 and NO2 abruptly increase, and so does the number of active IPs in both ASes. This strongly indicates that flows served by Akamai under normal operation (i.e., the majority of the time) are now served by neighboring ISPs. Akamai actually deploys servers inside the ISPs [6], which also explains the synchronized shift of flows. Fig. 4 depicts a 12 hours zoom around the events C and D. During the event C, the Akamai drop is again compensated by NO1 and NO2 in terms of volume. However, unlike NO2, there is a limited increase in the number of flows served from NO1, suggesting that the latter takes over the largest flows from Akamai. Event D differs from the previous ones since it does not involve Akamai, and it is characterized by a swap in the number of flows between NO1 and NO2.

We acknowledge that we do not know the ground truth or root causes causing the aforementioned unexpected cache selection events. We did not observe any abrupt variation in the total traffic, throughput, average RTT to the active IPs, nor in the number of erroneous HTTP responses during the events A-D, suggesting that the cache selection did not impact the end-user QoE. However, we argue that these fast and significant traffic shifts might be highly costly for the LO. Indeed, we verified via traceroutes that Akamai, NO1, and NO2 are neighbors to LO. As reported in the Internet AS-level topology archive[3], the relation between LO and Akamai is peer-to-peer (P2P), whereas the relation between LO and both NO1 and NO2 is customer-to-provider (C2P). In a nutshell, the P2P relation results in no transit costs for the LO for the flows served by Akamai, whereas the C2P relation might represent additional transit costs for the LO for flows coming from NO1 and NO2. For this reason, such events are worth to be automatically detected and analyzed.

A. *Temporal Similarities in Facebook and Akamai Traffic*

To get further insights on how to detect the aforementioned anomalies using our statistical approach, we investigate the temporal evolution of the probability distributions of the flow counts across IPs serving the Facebook content. The flow counts are computed for each observed server IP, considering different time-scales to enable multi-scale analysis (e.g., from 1' to 60'). The distribution of the flow counts across the server IPs is computed after each time bin. Finally, by comparing the distributions referring to different time intervals through the

[3]Internet AS-level Topology Archive, http://irl.cs.ucla.edu/topology/.

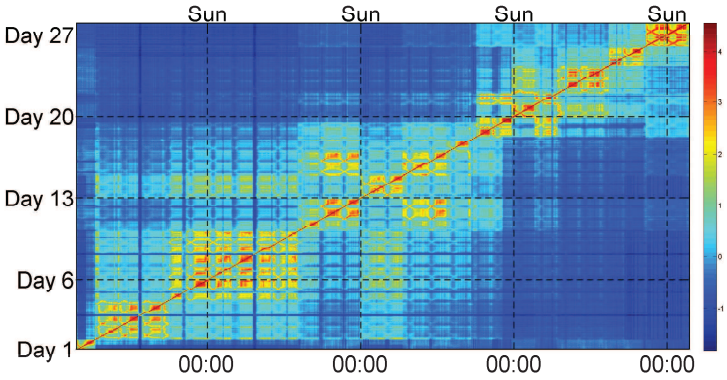


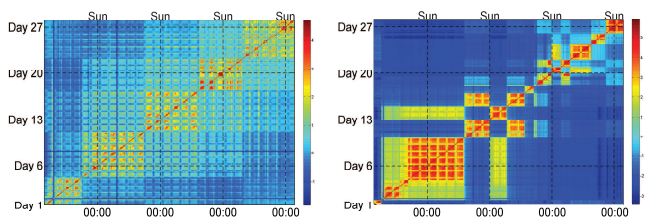
Figure 5. TSP of flow count distributions at 1h time scale, over 28 days.

modified K-L divergence (2), we get a direct insight on how the flow load balancing is performed among the IPs of the different organizations. 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 [7], 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.

Fig. 5 gives an example of a TSP for the distributions of all the Facebook flows across all the server IPs providing Facebook content, over the complete span of the dataset. The blue palette represents low similarity values, while reddish colors correspond to high similarity values. The TSP is symmetric around the 45° diagonal, thus the plot can be read either by column or by row. For a generic value of the ordinate at t_j, the points on the left (right) of the diagonal represent the degree of similarity between the past (future) distributions w.r.t. the reference distribution at t_j.

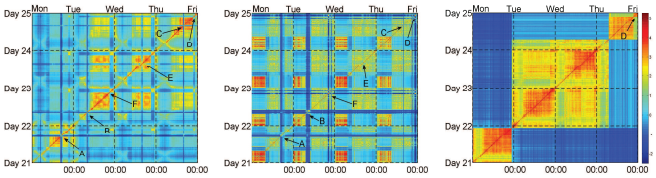
The TSP in Fig. 5 refers to the distributions on a time-scale of 1 hour. Note the regular "tile-wise" texture within a period of 24 hours, due to the daily cycle. The lighter zones correspond to the day-time periods, whereas the dark blue zones correspond to the night-time periods when the traffic load is low. The low similarity at night (02:00-05:00) is caused by the low number of flows, inducing larger statistical fluctuations. This pattern repeats almost identical for a few days, forming multi-days macro-blocks around the main diagonal, of size ranging from 2 up to 6 days. 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. Indeed, it discloses a reusage of (almost) the same address range between days 4-10 and 14-15, and between days 11-13 and 16-17. Finally, we observe a sharp discontinuity on days 19-20.

To better understand these behaviors, we separately plot the two main sources of Facebook flows, namely Akamai and the Facebook AS. Comparing Figs. 6(a) and 6(b) against Fig. 5 shows a very different allocation policy used by the two organizations. Akamai uses the same IPs for 4 to 7 days (see multi-day blocks around the main diagonal). When it changes the IPs the shift is not complete, as we can observe



(a) Akamai (b) Facebook AS

Figure 6. TSP of flow counts distributions at 1h time-scale.



(a) Akamai (b) NO1 (c) Facebook AS

Figure 7. TSP of flow counts distributions at 5' time-scale.

the macro-blocks slowly fading out over time. This suggests a rotation policy of the address space of Akamai on a time-scale longer than a month. However, we cannot prove this conjecture because of the limited duration of the analyzed dataset. Facebook AS does not reveal such a clear temporal allocation policy. It alternates periods of high stability (e.g. between days 4-10) with highly dynamic periods (e.g., from day 19 onward). Note that Facebook AS is responsible for the IP reuse between days 4-10 and 14-15, and between days 11-13 and 16-17, and for the abrupt change on days 19-20, both already identified in Fig. 5. Finally, NO1 always uses two distinct address sets during the night and the day periods, as depicted in Fig. 7(b).

We can use the TSPs to identify, by graphical inspection, the aforementioned anomalies in the traffic distributions. Indeed, a transient anomalous event appears in the TSP as a full blue cross centered on the main diagonal, at the time of the event. Fig. 7 shows the TSPs of the flow counts distributions between days 21 and 24 at a 5 minutes time-scale (i.e., the same period and aggregation depicted in Fig. 3), for Akamai, NO1, and Facebook AS respectively. The events A, B, and C are clearly visible in the TSPs of Akamai and NO1, and are totally absent from the Facebook AS TSP. These events are also clearly visible in the TSP of NO2 (not reported for space limitations), and are in total accordance with the analysis for the flow counts time-series in Figs. 3 and 4. Regarding the event D, it is observable in all the TSPs, even though it is completely invisible in the time-series of flow counts and volume of Facebook AS in Fig. 4. Furthermore, Figs. 7(b) and 7(a) pinpoint the presence of two more anomalous events in the Akamai and NO1 traffic, namely the events E and F, that are completely invisible in the flow and volume plots. This additionally justifies the usage of probability distribution based approaches for detecting such abnormal events.

B. Detecting Service Outages in Facebook

To conclude with the analysis of anomalies in Facebook traffic, we devote the last part of this section to the detection

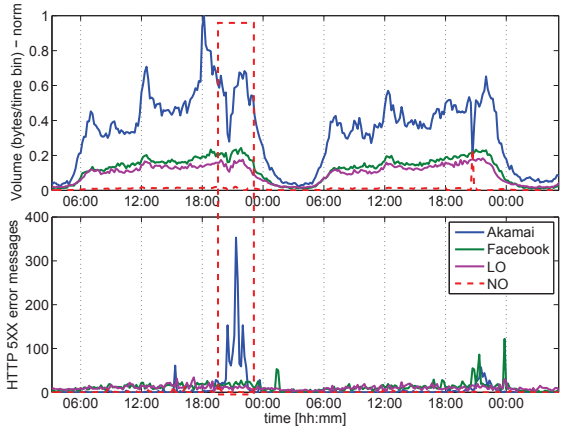


Figure 8. Detection of Facebook outages in Septmeber 2013. (up) Facebook downlink traffic volume per AS and (down) HTTP server error message (e.g. 5XX) counts.

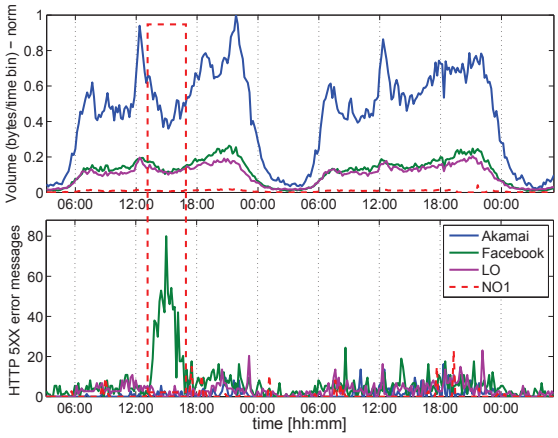


Figure 9. Detection of Facebook outages in October 2013. (up) Facebook downlink traffic volume per AS and (down) HTTP server error message (e.g. 5XX) counts.

of outages in the Facebook service. Such outages are not directly linked to the cache selection policies employed by the CDNs serving the content, but are still related to them, as they may occur at different ASes hosting the service. Fig. 8 depicts a very interesting event detected by our approach in the Facebook traffic served by Akamai, which we claim corresponds to a large outage in Akamai servers during a time frame of about 2 hours in September 2013. The total volume served by Akamai, Facebook AS and LO abruptly drops during this outage, being Akamai the organization showing the highest change. Different from the events previously analyzed in Figs. 3 and 4, no other organization takes over the dropped traffic, suggesting the occurrence of an outage.

To further understand the root causes of the abrupt drop, Fig. 8 additionally plots the time series of the count of HTTP server error messages (i.e., 5XX HTTP answers) corresponding to the Facebook HTTP flows served by the aforementioned ASes. The high increase in the counts for Akamai is impressive, meaning that during the volume drop, the HTTP web traffic hosted by Akamai was not available for many of users. The increase of the 5XX messages continues for about half an hour after the apparent recovery, flagging some transient effects which might be linked to the re-start of some servers. Interestingly, there are no noticeable variations in the counts for the other ASes, suggesting that the outage is only part of the Akamai CDN and is not related to the Facebook service itself. As we said before, we do not have any ground truth flagging this outage in the Akamai CDN. However, we also detected an outage of very similar characteristics about one month later, for which we have the ground truth of its occurrence, disclosed in the international press[4].

Fig. 9 depicts this new outage occurring in October 2013. The drop in the served volume is not as marked as before, and in this case, the increase in the HTTP error message counts occurs for the servers under Facebook AS and not Akamai. However, the characteristics are very similar: a drop in the overall served volume with no other organization taking over, as well as a marked increase in the HTTP error messages counts. According to the press release, this Facebook outage was caused by maintenance issues. As a final statement on the importance of rapidly detecting and diagnosing these types of events we cite directly the press release, which claims that the flagged outage impacted millions of Facebook users on more than 3.000 domains. Interestingly for IPSs, the experts behind the press release advise to check the status of large services like Facebook before actually starting a troubleshooting phase on their internal systems.

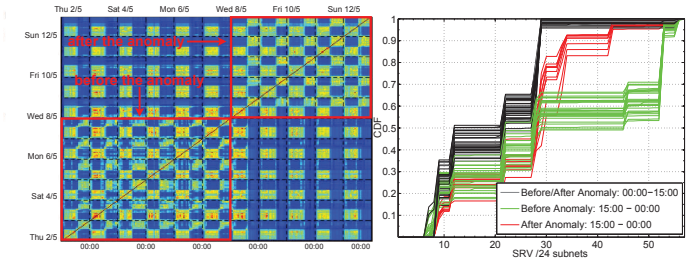
[4]http://www.theguardian.com/technology/2013/oct/21/facebook-problems-status-updates

V. User Experience Anomalies in YouTube

CDN cache selection policies may also 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 results in an important drop on the average download throughput for the end-users watching YouTube videos. Indeed, 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 caches selection done by Google, the ISP internal RCA did not identify any problems inside its boundaries. As we said in the last part of previous section, this standard procedure followed by operators should always be complemented with a verification of the status of the services being accessed by the users, which in many cases are the root of the problems.

The 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



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

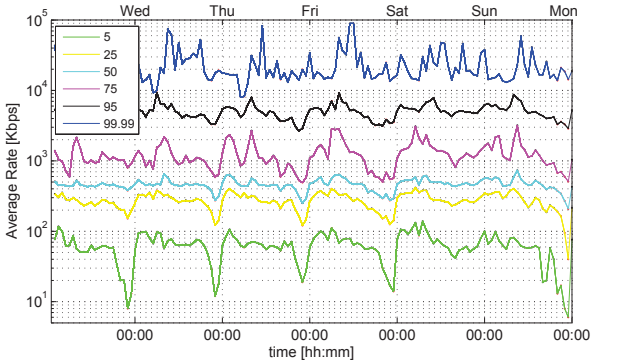
Figure 10. Traffic volume distributions per CDN /24 subnets.

Tstat passive monitoring system [14]. Using Tstat filtering and classification modules, we only keep those flows carrying YouTube videos. These flows are finally imported and analyzed with DBStream.

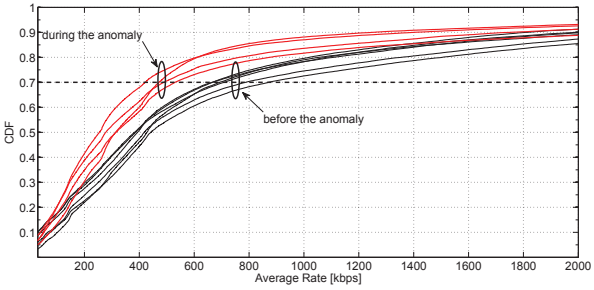
As reported by the ISP operations team, the anomaly occurs on Wednesday the 8th of May. Fig. 10(a) shows the TSP of the video volume served by the different IPs in the dataset, aggregated in /24 subnetworks, and using a time-scale of 1 hour. Similar to the Akamai case, we can appreciate a marked daily periodicity behavior in the TSP. Specifically, there are two subnet sets periodically re-used in the first and second half of the day. The TSP clearly reveals that a different subnet set is used during the second half of the day from the 8th of May on, revealing a different cache selection policy. This change is also visible in the CDFs of the per subnet volume depicted in Fig. 10(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.

Despite this detected change in the cache selection policy employed by Google, such a modification does not justify by itself the QoE degradation reported by the ISP. To further investigate this issue, we analyze the distributions of the average video flows download rate. Fig. 11(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, explaining the flagged QoE degradations. Fig. 11(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 Fig. 11(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 [16] that it is sufficient to drop the perceived quality below the level of acceptance.

Fig. 12 permits to better explain the QoE degradation. The Fig. reports the overall QoE and the acceptance rate as declared by users watching YouTube videos during a field trial test conducted and reported in [16], both as a function of



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

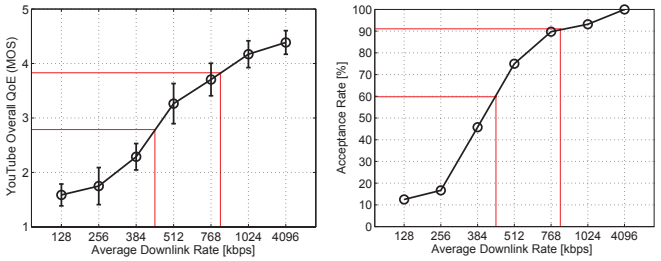


(b) CDFs before and during the anomaly.

Figure 11. 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.

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 evaluated anomalous situation, not only 360p videos were consumed by the users, but most probably videos with higher resolutions (e.g., 1080p HD), and thus we expect that the impacts on the user experience were even more severe than what we report in here.

Fig. 12(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 Fig. 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. Fig. 12(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 impacts of such downlink rate drop on the users. To conclude the analysis, we report in Fig. 13 the output of the proposed AD system. Fig. 13(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



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

Figure 12. 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.

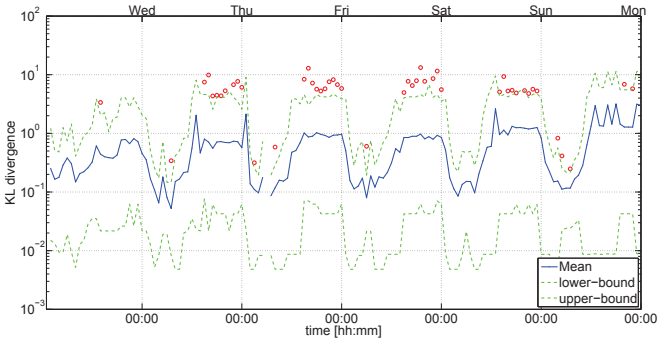
changes. The red markers indicate when the condition $\Gamma(t) < \Phi_\alpha(t)$ is violated, triggering an anomaly. From Wednesday the 8th of May onward the algorithm systematically rises alarms from 15:00 to 00:00, which correspond to the discussed change in the caching policy. Fig. 13(b) reports 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 Fig. 11. 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 Fig. 1(a). Indeed, the percentiles depicted in Fig. 11(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 a 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.

VI. Discussion and Concluding Remarks

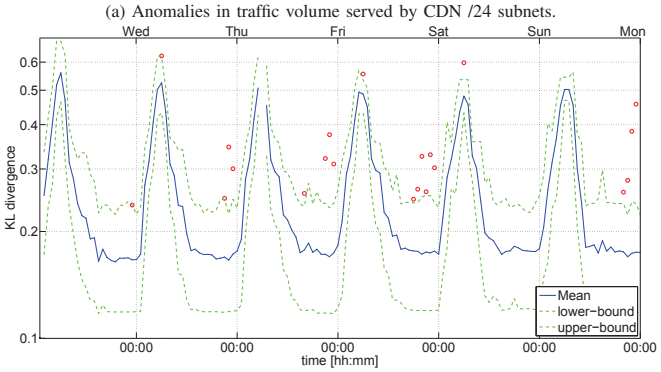
In this paper, we have shown that the caching selection policies employed by major CDNs might have an important impact on both the ISP carrying the traffic and the end-customers. Our study considered traffic from two large and very different datasets collected in different countries, showing that these events are not bound to a particular location or type of network. We argue that it is important for the ISP to rapidly and automatically detect the occurrence of such events, and therefore presented a network AD system for CDNs' traffic. By applying this algorithm to the traffic datasets, we were able to identify different classes of anomalies in Akamai and Google CDN. In this paper we have not fully evaluated the limitations of our AD system to cope with the high complexity of the considered scenarios, which is part of our ongoing work. The algorithm is currently running in the wild in two different operational networks, with the aim of detecting and characterizing the types of anomalies typically observed in CDN-provisioned services.

Acknowledgments

This work has been partially funded by the EU-IP project mPlane under grant 318627, and by the Darwin4 project at the Telecommunications Research Center Vienna.



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



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

Figure 13. 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.

References

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

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

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

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

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

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

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

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

[9] M. Stoecklin, "Anomaly Detection by Finding Feature Distribution Outliers", in *CoNEXT*, 2006.

[10] Y. Gu et al., "Detecting Anomalies in Network Traffic using Maximum Entropy Estimation", in *IMC*, 2005.

[11] T. Dasu et al., "An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams", in *INTERFACE*, 2006.

[12] P. Fiadino et al., "HTTPTag: A Flexible On-line HTTP Classification System for Operational 3G Networks", in *INFOCOM*, 2013.

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

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

[15] P. Romirer et al., "On the Use of TCP Passive Measurements for Anomaly Detection: A Case Study from an Operational 3G Network", in *TMA*, 2010.

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

[17] A. D'Alconzo et al., "Who to Blame when YouTube is not Working? Detecting Anomalies in CDN-Provisioned Services", in *TRAC*, 2014.

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