Towards Automatic Detection and Diagnosis of Internet Service Anomalies via DNS Traffic Analysis

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

Abstract—**The DNS protocol has proved to be a valuable means for identifying and dissecting large-scale anomalies in omnipresent Over The Top (OTT) Internet services. In this paper, we present and evaluate a framework for detecting and diagnosing traffic anomalies via DNS traffic analysis. Detection of such anomalies is achieved by monitoring different DNS-related *symptomatic features*, flagging a warning as soon as one or more of them show a significant change. The investigation of the root causes for such deviations is done by looking at significant changes in a number of *diagnostic features* (i.e., device manufacturer and OS, requested host name, error codes, etc.), which convey information directly linked to the potential origins of the detected anomalies. For the purpose of detecting significant changes in the time-series of diagnostic features, we propose a scheme based on change point detection applied to the entropy of the considered features. The proposed solution is tested using both real and synthetic data from a nationwide mobile ISP, the latter generated from real traffic statistics to resemble the real mobile network traffic. To show the operational value of the proposed framework, we report the results of the diagnosis in two prototypical cases.**

Keywords—Anomaly Detection; RCA; Network Measurements; Statistical Analysis.

I. Introduction

During the last decade, a plethora of new, heterogeneous Internet-services have become highly popular and omnipresent, imposing new challenges to network operators. The complex provisioning systems used by these services induce continuous changes that impact both operators and customers. Indeed, efficient traffic engineering becomes a moving target for the operator [1], and management of Quality of Experience (QoE) gets more cumbersome, potentially affecting the end customer [3]. Furthermore, due to their traffic characteristics, applications that provide continuous online presence (e.g., messaging services) might severely impact the signaling plane of the network, especially in mobile networks [6]. In such a complex scenario, it is of vital importance to promptly detect and diagnose the occurrence of large scale changes that could result in anomalies for some of the involved stakeholders.

In this paper we take a step forward from our previous work [4], where we have initially investigated the problem of diagnosing network traffic anomalies caused by specific devices and applications. Here we propose a more general detection and diagnosis framework, and cast it to the analysis of Domain Name System (DNS) traffic. The DNS is the core component of the Internet, providing a flexible decoupling of a service's domain name and the hosting IP addresses. Modern load balancing mechanisms rely on the diversification of DNS answers to different clusters of users [2], where the Time to Live (TTL) of those answers is usually short, in the order of a few seconds, allowing to quickly re-map hostnames and IP addresses as needed. As a consequence, every time a user tries to access a remote service, it is likely to generate a new DNS query. Based on these observations, anomalies in such services are likely to induce changes in the normal DNS usage patterns. For example, users accessing a temporary unreachable service would generate a new query at every connection retry [7].

Along with the DNS request-response transaction data, our approach assumes the availability of *meta-data*. In the specific case of traffic originated from a mobile network, these meta-data include information related to the end-host (e.g., device manufacturer, Operative System), the access network (e.g, Radio Access Type – RAT, Access Point Name – APN, IP address of the DNS resolver), and the requested service (e.g., requested Fully Qualified Domain Name – FQDN). Leveraging these DNS data, we extract two sets of features denoted as *symptomatic features* and *diagnostic features*. Symptomatic features are defined such that their abrupt change directly relates to the presence of abnormal and potentially harmful events, while diagnostic features shall provide contextual details of the anomalies, pointing to their root causes. Features are further processed to define what we shall refer to as analysis *signals*. Signals describe the statistical content of features, and allow for abstraction and generalization of the framework's input definition. For example, a relevant feature used in our framework is the number of DNS requests per observed FQDN in a certain time bin; in this case, a signal associated to this feature could be defined as the mean number of DNS requests, the total number, the full empirical distribution, the entropy, etc. Two signals derived from the same feature might yield completely different detection results: for example, an anomaly could be easily spotted when analyzing the entropy of a certain feature, but not through its mean value. The separation between feature and signal allows to decouple the meaning of an input from the information it exposes for detecting anomalies.

The contributions of this paper are as follows: (i) a generic detection and diagnosis framework for Internet service anomalies, (ii) a procedure to generate semi-synthetic DNS traffic and to model different types of anomalies observed in real mobile traffic, and (iii) an initial evaluation of the proposed framework, using an entropy-based detection scheme.

The remainder of this paper is organized as follows: Sec. II overviews the proposed diagnosis framework. Sec. III describes the change detection approach used to pinpoint changes in the DNS traffic. Sec. IV is devoted to the characterization of the DNS traffic and the generation of synthetic datasets. In Sec.

978-1-4799-5344-8/15/$31.00 ©2015 IEEE

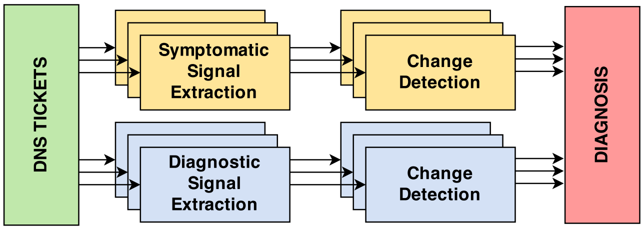

Figure 1. Overview of the detection and diagnosis framework.

V we discuss the results obtained by applying the framework to two specific types of anomalies, modeled from the real mobile network traffic. Finally, Sec. VI concludes this work.

II. Detection and Diagnosis Framework

In this section we describe a generic framework to detect and diagnose large-scale network traffic anomalies, applied to the analysis of DNS traffic. Fig. 1 sketches an overview of the proposed framework in this application scenario. From DNS tickets we derive two sets of signals denoted as *symptomatic signals* and *diagnostic signals*. All signals are checked for significant changes from their reference of "normality". However, the symptomatic signals are designed such that their changes directly relate to the presence of abnormal and potentially harmful events. On the other hand, changes in the diagnostic signals per-se do not have a negative connotation, but rather ease and guide the interpretation of the anomalous event. In the diagnosis step, deviations of the symptomatic signals are correlated with the subset of simultaneously changing diagnostic signals to provide a comprehensive characterization of the event.

We assume that a network monitoring system tracks the DNS traffic at a vantage point in the core of a typical mobile network. A DNS ticket summarizes info related to a DNS query generated by a mobile device, and the reply from the DNS resolvers. The tickets also contain the meta-data listed in Tab. I. These data provide further information about the mobile device (ID, manufacturer, OS), the network settings (APN, RAT, DNS server IP), and the FQDNs of the remote service. Additionally, the Error Flag records the status of the DNS transaction as tracked by the network monitoring system (i.e., successful, time-out, retransmission).

As abrupt changes in the number of DNS queries often indicate the presence of application- or device-specific anomalies, the number of DNS queries per unit of time is amenable as symptomatic feature. As for the diagnostic signals, we look at the distribution of the fields in Tab. I to characterize the sub-population of devices affected by the anomaly (manufacturer, OS), the remote service involved (FQDN), the network settings (APN, server IP), and finally the possible effect on the DNS infrastructure (Error Flag). In this work we focus exclusively on DNS-related signals. However, the proposed framework is generic enough to operate with different network data sources, such as flow-level or application layer information, which might contribute to further refine the anomaly diagnosis. We leave this for future work.

The output of the change detection feeds the final *Diagnosis* block of the framework, whose formal definition is part of our ongoing work. We envision a smart module capable of finding temporal correlations of signals' changes. The framework should be ultimately able to automatically generate an event fingerprint in correspondence to one or more anomaly symptoms, enriched with diagnostic information. Such a report would provide a valuable basis for supporting the anomaly troubleshooting process.

Field Name	Description
Device ID	Anonymized device identifier
Manufacturer	Device manufacturer
OS	Device operating system
APN	Access Point Name
RAT	Radio Access Type
DNS Server	IP address of the DNS resolver
FQDN	Fully Qualified Domain Name of remote service
Error Flag	Status of the DNS transaction

Table I. DNS ticket information (meta-data).

In this paper we focus on the two fundamental aspects of the proposed framework, namely the definition and extraction of the symptomatic and diagnostic signals, and the design of suitable detection schemes.

III. Signals Extraction and Change Detection

Let us now formalize the definition of features and signals as considered in the proposed framework, as well as describe the applied change detection technique.

For the generic feature f derived from the DNS tickets, we indicate by $f_i^\tau(t)$ the generic counter observed at the t-th time bin of length τ. For instance, if f represents the number of DNS requests for a FQDN every τ minutes, $i \in \{1, \ldots, n(t)\}$ is the i-th requested FQDN, while $f_i^\tau(t)$ counts the number of DNS requests for the i-th FQDN (out of the $n(t)$), over the t-th time bin. The i-th counter can also be associated to other fields, such as the OS version, the Error Flag value, the number of DNS queries generated by the i-th device, etc. The length of τ defines the timescale of the data aggregation, which in turn defines the timescale of the observable anomalous events. 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 $X_f^\tau(t)$. By properly grouping features, we can obtain aggregated statistics for different "views" on the data. Considering the example above, the FQDN counters can be further grouped to obtain, e.g., 2-nd Level Domain (2LD) or 3rd Level Domain (3LD) counters and statistics. We can also use the counters for computing the overall number of DNS requests as $N(t) = \sum_i f_i^\tau(t)$.

Additionally, from the empirical distribution of the feature, $X_f^\tau(t)$, we can compute its normalized entropy as:

$$H(X) = -\frac{1}{\log(|\Omega|)} \sum_{\omega \in \Omega} x(\omega) \log x(\omega), \qquad (1)$$

where Ω and $|\Omega|$ are a discrete probability space and its cardinality, respectively. As the following analysis can be done independently of the specific selected time scale, we omit the superscript τ from now on.

In this paper, we consider the total number of queries $N(\tau)$ as the single symptomatic signal of the framework. Since entropy is a well-known synthetic index for describing an entire distribution [8], we consider the normalized entropy

$H(X)$ of the features in Tab. I as diagnostic signals. Entropy-based approaches have been proposed for traffic analysis in the past [10], but in a fixed-line network context, and using only standard transport-layer features such as IPs and ports. Our analysis specifically targets cellular networks, using a much richer set of features, from network to device specific ones.

The change detection scheme we consider is based on the well-known Exponential Weighted Moving Average (EWMA) algorithm. EWMA weights past observations such that the older ones count less in the determination of the expected current value. For the observed value $y(t)$ at time bin t, the value predicted by EWMA is calculated as:

$$\tilde{y}(t) = \lambda y(t) + (1 - \lambda)\tilde{y}(t-1), \tag{2}$$

where λ controls the filter *memory*, that is the weight of the past samples in computing the moving average; the higher λ, the higher the weight of the newer samples. For detection purposes, the acceptance region for normality is defined by an Upper Control Limit (U_{CL}) and a Lower Control Limit (L_{CL}), defined as:

$$U_{CL}(t) = (1 + \sigma)\tilde{y}(t-1) \tag{3}$$

$$L_{CL}(t) = (1 - \sigma)\tilde{y}(t-1) \tag{4}$$

The parameter σ is a slack factor that controls the width of the acceptance region for normality. Finally, the detection algorithm flags an anomaly if $y(t) \notin [L_{CL}(t), U_{CL}(t)]$. Note that by opportunely tuning λ and σ, the EWMA-based detection algorithm becomes able to accommodate for typical daily and weekly patterns of real network traffic.

IV. DNS Traffic Characterization and Anomaly Modeling

We evaluated the proposed framework for longer than six months in 2014 with traffic from the operational cellular network of a nationwide European operator. The extensive experimentation allowed us to collect results in a number of paradigmatic case-studies exposing features and limitations of the framework. Still, the number of DNS traffic anomalies observed in the corresponding period was relatively low, limiting as such the chances of performing a complete performance analysis of the framework by relying exclusively on real traffic. In principle, one could resort to test traces obtained in a controlled environment (laboratory) or by simulations, but these approaches would miss the complexity and heterogeneity of the real traffic.

To bypass this hurdle, we adopted a methodology based on semi-synthetic data, derived from real traffic traces as suggested in [11]. Such an approach does not only allows to extensively analyze the performance of the framework with a large number of synthetic, yet statistically relevant anomalies, but also permits to protect the operator's business sensitive information, as neither real data traces nor real anomalies are exposed. Next we explain the procedures for both generating a semi-synthetic background DNS traffic, as well as for modeling the DNS-related anomalies for replicating them.

A. Semi-synthetic Dataset Construction

The procedure for constructing the semi-synthetic dataset is conceived with the objective of maintaining as much as possible the structural characteristics of the real, normal operation (i.e., anomaly-free) traffic, while eliminating possible (unknown) anomalies present in real traces. Exploring real traces, we observed that the traffic yields some fundamental temporal characteristics. In particular, the traffic is non-stationary due to time-of-day variations. This effect is not limited to the number of active devices and the number of DNS queries, but rather applies to the entire distribution. Distribution variations depend on the change of the applications and terminals mix, which in turn induce modifications in the DNS request generation patterns. Furthermore, we found that, besides a strong 24-hours seasonality, the DNS traffic exhibits a weekly pseudo-cyclicity with marked differences between working days and weekends/festivities [7]. Finally, traffic remains pretty similar at the same time of day across days of the same type.

The first step of the construction procedure consists of manually labeling and removing possible anomalous events. However, as the complete ground truth is unknown in real traffic, we cannot completely rely on individual labeling of alarms. Therefore, we have to accept that minor anomalies may go undetected if their effect is comparable with purely random fluctuations. Then, the dataset is transformed to eliminate possible residual (unknown) anomalies present in the real traffic, while preserving the above mentioned structural characteristics. The transformation procedure is described as follows.

Let consider a real dataset spanning a measurement period of a few weeks, for a total of m consecutive one-day intervals (e.g., $m = 28$ in our case). Each one-day period starts and ends at 4:00 am local time: this is the time-of-day where the number of active devices reaches its minimum (considering a single time-zone). Denote by m_W and m_F the number of working and festivity (W- and F-) days, respectively, in the real dataset (e.g., $m_F = 8$ and $m_W = 20$), and by K the total number of 1-min timebins ($K = 28{\cdot}24{\cdot}60 = 40320$). For each device i consider the vector $\mathbf{d}_i \equiv \{c_i^{\tau_0}(k), k = 1, 2, \ldots, K\}$ at the minimum timescale ($\tau_0 = 1$ minute) across the whole real trace duration, where each element $c_i^{\tau_0}(k)$ is the list of the DNS tickets related to device i at time k. For those timebins where device i is inactive, the corresponding element in $\mathbf{d}_i$ is empty. We now divide this vector into m blocks, each one corresponding to a single one-day interval. Each block is classified as W- or F-block based on the calendar day. At this point we apply a random *scrambling* within the W class: each W-block element of $\mathbf{d}_i$ is randomly relocated at the same time position selected among all W-days. The same scrambling is applied independently to the F-blocks. In this way we obtain a new vector $\widetilde{\mathbf{d}}_i$ where the position of the blocks has been scrambled, separately for W- and F-blocks, but the time location and the F/W intervals have been maintained. Finally, from the set of scrambled vectors $\widetilde{\mathbf{d}}_i$ we can derive a new set of distributions for each timebin k and timescale τ, for all the considered traffic features.

The dataset obtained in this way retains certain characteristics of the real dataset, while others are eliminated. The most important change is that the random scrambling of the individual components $\mathbf{d}_i \rightarrow \widetilde{\mathbf{d}}_i$ results in the *homogenization* of the individual daily profiles — separately for W- and F-days. This eliminates any residual local anomaly by spreading it out across all one-day intervals of the same F/W type.

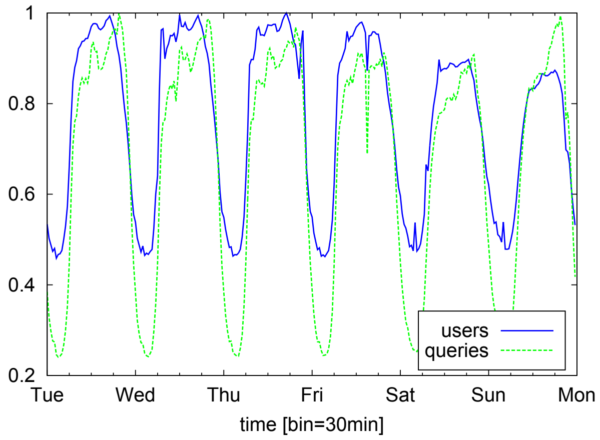

Figure 2. Daily trend of the number of active users and total DNS query count in the semi-synthetic dataset.

In other words, all W-days in the new dataset share the same (synthetic) aggregate daily profile. Same applies to F-days. Note however that the synthetic dataset retains the most important characteristics of the real process. In the first place, it keeps the time-of-day variations of the number of active devices (see Fig. 2). However, the total number of queries at time k changes as permuted devices issue (in general) different amount of DNS queries. Secondly, the semi-synthetic dataset maintains the differentiation between the two classes of W- and F- days, although it eliminates any differentiation *within* each class (e.g., between Saturday and Sunday). Thirdly, it keeps the differentiation between distributions for different time-of-day. This is clear from Fig. 3, which shows the hourly Cumulative Distribution Functions (CDFs) of the number of devices across query count during one day of the semi-synthetic dataset. The result of the procedure is an anomaly-free DNS dataset *structurally similar* to the real trace.

B. Construction of Synthetic Anomalies

During six months of experimentation we encountered a few recurring large-scale DNS traffic anomalies. Investigating these events we found some common traits and we conceived a procedure for reproducing them along with their most relevant characteristics. In particular, we identified two exemplary event types (E_1 and E_2 from now). In both the cases, we model an outage of an Internet service for a specific sub-population of devices, which react by repeatedly and constantly issuing DNS queries to resolve the requested service throughout the anomaly. Involved devices are identified by fixing a specific OS (with its different versions). Moreover, we aim at modeling the correlation between the selected sub-population and the unreachable service. Therefore, we separately rank the 2LDs of the FQDNs for anomalous and background traffic, and select the most popular 2LD of the former that is not in the latter.

The event of type E_1 models the case of a short lived (i.e., hours) high intensity anomaly (e.g., 10% of devices repeating a request every few seconds), where all the involved devices are produced by a single manufacturer and run the same OS. In this case, the number of involved terminals and the overall number of additional queries is such to overload the local DNS servers. The latter effect is modeled by increasing the number of `time-out` codes in the Error Flag field.

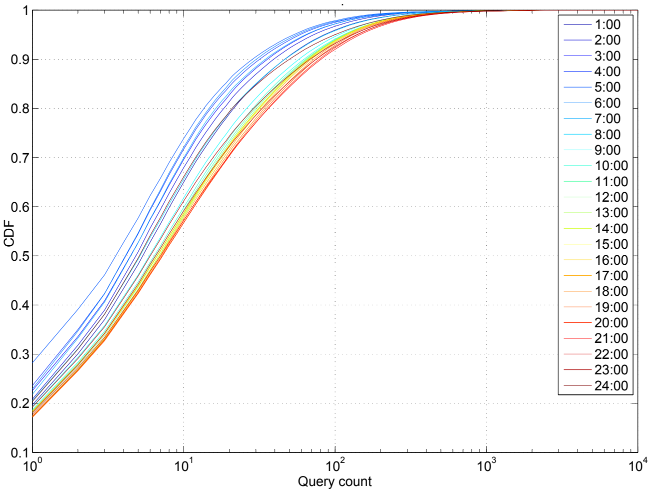

Figure 3. Hourly trend of the distribution of number of devices across query count over one day of the semi-synthetic dataset.

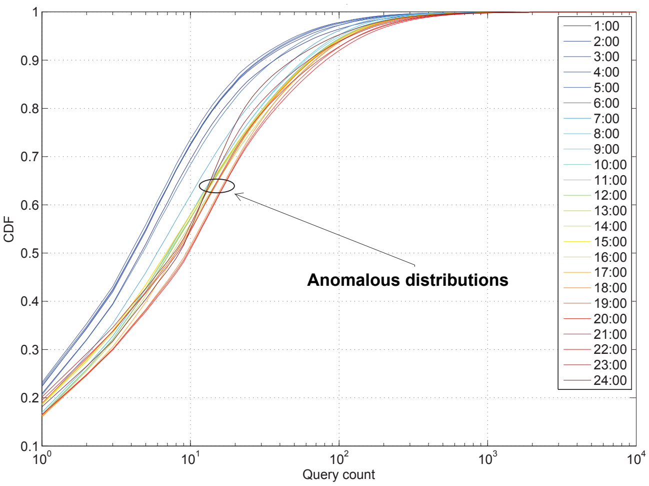

Figure 4. Hourly trend of the distribution of number of devices across query count over the first day of the E_2 anomaly.

The event of type E_2 models a long lasting (i.e., days) low-intensity anomaly (e.g., 5% of devices repeating requests every few minutes). Differently from the previous case, the involved terminals are produced by multiple manufacturers, even if they share the same OS. Given the low-intensity, we did not introduce a modification in the distribution of the Error Flag. Fig. 4 shows the changes in the distribution of number of devices across query counts introduced by this event (cfr. Fig. 3). Note that although E_2 type anomalies are of relatively low intensity, their identification is important as, in our experience, they may lead to problems on the signaling plane, such as resources starvation at the radio access.

Tab. II summarizes the characteristics of the two event types and the actual values used for generating the anomalous ticket dataset in the experiments discussed in Sec. V. To illustrate the anomaly generation procedure, we consider an event of type E_1 of duration $d = 1h$, starting at $t_1 = 9:00$.

Type	E_1	E_2
Start time t_1	9:00	13:00
Duration d	1h	2 days
Involved devices D	10%	5%
Back-off time	5 sec	180 sec
Manufacturer	single popular	multiple
OS	single (with sub-versions)	single (with sub-versions)
Error flag	+5% timeout	—
FQDN	top-2LD for involved devices	top-2LD for involvede devices

Table II. CHARACTERISTICS OF THE ANOMALOUS DNS TRAFFIC FOR TYPE E_1 AND E_2.

Starting from t_1 at each time-bin, $D = 10\%$ of all the active terminals are randomly extracted from the semi-synthetic background traffic, such that the OS is the selected one and the manufacturer is always the same. For each involved terminal, we generate one additional DNS ticket every 5 seconds, which are then added to the semi-synthetic dataset. The FQDN in these tickets is randomly chosen among the domains in the 2LD identified as explained above. Finally, the Error Flag is changed to `time-out` in 5% of the overall DNS tickets, so as to model the resolver overload. The last step consists of mangling both the anomalous and the background traffic.

The procedure for generating type E_2 is analogous, but differs in the selection of the anomalous terminals (same OS, but not necessarily same manufacturer). The Error Flag is not changed.

V. EVALUATION AND RESULTS

In this section we present the results on the performance evaluation of the framework in terms of detection and diagnosis capabilities, for the case of the aforementioned anomalies of E_1 and E_2 type.

A. Anomalous Event of Type E_1

As described in Sec. IV, the first event is characterized by a short duration (1 hour) and a high intensity, as it involves a large population of devices (10%) coming from the same popular manufacturer and running one specific OS. The evaluation is performed at a $\tau = 5$ minutes time scale.

Fig. 5 shows the time-series of the DNS query count, used as symptomatic signal. The gray area highlights the event time span, from 9am to 10am. The increase on the number of DNS queries is clearly visible, resulting in about the double of the number of queries as observed in normal operation conditions at that time of the day. The red points in the figure indicate the deviations flagged by the EWMA algorithm. The entropy trend of the diagnostic features is depicted in Fig. 6. Given the high intensity of the event, marked variations are visible in all the diagnostic signals. In fact, the fraction of DNS queries generated by devices with a specific manufacturer and OS changes during the event, hence the entropy of the respective dimensions exhibit a sharp decrease. Similarly, the FQDN entropy signal decreases, since the affected devices repeatedly try to contact a specific service. On the contrary, the Error Flag diagnostic signal shows a significant increase. In fact, the increased share of `time-outed` queries perturbs the distribution else concentrated around the `successful` value, with a consequent spike in the entropy value. The notches and spikes in the entropy, as well as in the query count, are easily detected by the EWMA algorithm.

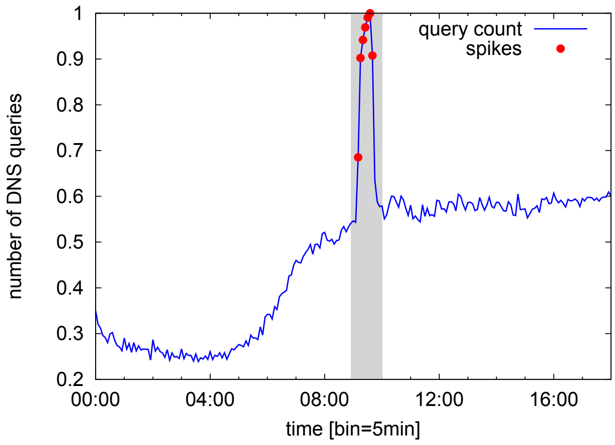

Figure 5. EWMA change point detector applied to symptomatic signal (query count) in event type E_1. The event is highlighted in the gray area. The red dots marked as *spikes* correspond to the alarms flagged by the detector.

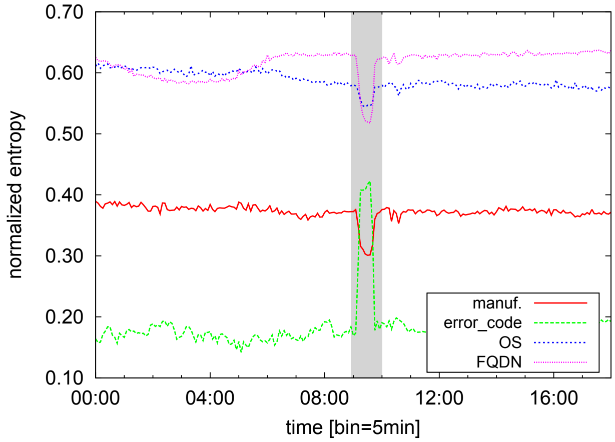

Figure 6. Normalized entropy of diagnostic features in event type E_1. All signals are clearly altered (spikes and notches) during the event.

Summarizing the first experiment on E_1, the framework allows to accurately detect the changes on the symptomatic signal, as well as on the diagnostic signals. The changes on all signals are simultaneously detected, providing a reliable input to the diagnosis step.

B. Anomalous Event of Type E_2

Differently from event E_1, the anomalous event of type E_2 involves a smaller population of devices (5%) running an OS pre-installed by a number of different manufacturers. Similarly to the previous case, the affected terminals continuously try to re-contact the servers hosting an unreachable service. Because of the low intensity and the longer duration of the event, the analysis is performed at $\tau = 30$ minutes time scale.

Fig. 7 depicts the time-series of the query count during a period of 3 days, which includes the anomalous event, starting at 1pm of the first day and lasting till 11am of the third day. The counter shows a slight increase during the anomaly, but the EWMA detection algorithm only flags changes at the

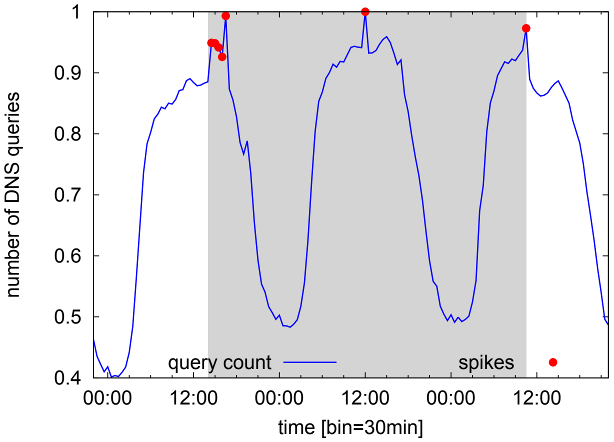

Figure 7. EWMA change point detector applied to symptomatic signal (query count) in event type E_2. The event is highlighted in the gray area. The red dots marked as *spikes* correspond to the alarms flagged by the detector.

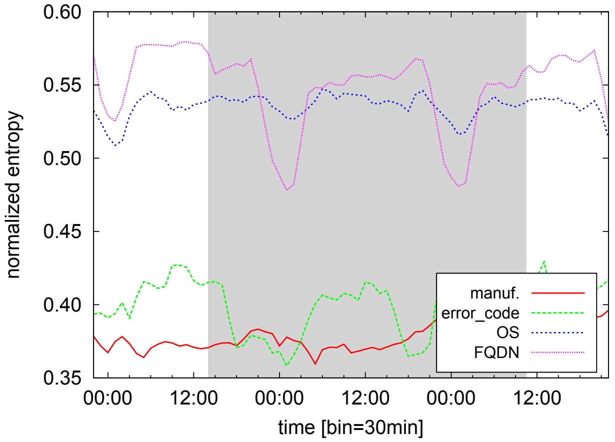

Figure 8. Normalized entropy of the diagnostic features in E_2 type event. No clear evidence of the underlying event, with the exceptions of two notches in the FQDN signal.

beginning, and is not able to track the anomaly during its complete time span.

Fig. 8 plots the trend of the diagnostic signals. Only the FQDN entropy exhibits evident changes during the night-time, when the increased number of queries towards the affected service stems out from the background night traffic. For the rest of the diagnostic signals, it is hard to claim that the small, low-speed observed changes could be detected, specially as they look very similarly to the patterns observed during normal operation. Indeed, the EWMA algorithm fails to track the full dynamics of the event. For what concerns the OS, the changes in the distribution induced by E_2 are not sufficient to alter the entropy signal. We recall that E_2 does not affect the Error Flag signal by design.

Summarizing the second experiment on E_2, using entropy-based diagnostic signals in the case of low intensity anomalies is not a viable solution, as the framework fails to reveal the effects of the anomaly in the diagnostic features. This suggests that other types of signals have to be considered for obtaining useful results in a more generic scenario, where anomalies' intensity would vary as well as the shifts in the specific diagnostic features. We are actually working in this direction now.

VI. Conclusions and Future Work

In this paper we have shown a preliminary study on a framework for automatic detection and diagnosis of large scale Internet anomalies based on DNS traffic. Its key idea is to apply a change point detection algorithm to a set of signals extracted from network measurements. To this extent, we have thoroughly tested it both on real network traces and semi-synthetic datasets. Despite initial promising results, the results presented in this paper unveiled some of the limitations of this approach. In particular, our change detection scheme has proved to be unreliable in the presence of low intensity anomalies that involve relatively small sub-populations of users, which are frequent in operational mobile networks, and still far from being innocuous (cfr. problems on the signaling plane).

Taking this into account, our current work is aimed at overcoming the limitations introduced by using entropy-based signals. The plan is to improve the diagnosis framework with a more complex change detection scheme that relies on the entire probability distributions rather than the entropy values, based or our previous work [5], [9]. By that, the system should be able to cope with anomalies that involve multiple services and/or affect multiple devices at the same time. Another objective of our future work is the design of a smart diagnosis module that correlates changes in symptomatic and diagnostic signals and automatically generates event fingerprints.

Acknowledgements

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627, "mPlane".

References

[1] P. Fiadino, A. D'Alconzo, P. Casas, "Characterizing Web Services Provisioning via CDNs: The Case of Facebook", in *TRAC*, 2014.

[2] P. Membrey, E. Plugge, D. Hows, "Practical Load Balancing: Ride the Performance Tiger", *APRESS*, 2012.

[3] P. Casas, A. D'Alconzo, P. Fiadino, A, Bär, A. Finamore, T. Zseby, "When YouTube doesn't Work – Analysis of QoE-relevant Degradation in Google CDN Traffic", *IEEE Transactions on Network and Service Managemen*, vol. 11, no. 4, 2014.

[4] M. Schiavone, P. Romirer-Maierhofer, P. Fiadino, P. Casas, "Diagnosing Device-Specific Anomalies in Cellular Networks", in *ACM CoNEXT Student Workshop*, 2014.

[5] P. Fiadino, A. D'Alconzo, A. Bär, A. Finamore, P. Casas, "On the Detection of Network Traffic Anomalies in Content Delivery Network Services", in *ITC*, 2014.

[6] A. Aucinas, N. Vallina-Rodriguez, Y. Grunenberger, V. Erramilli, K. Papagiannaki, J. Crowcroft, D. Wetherall, "Staying Online While Mobile: The Hidden Costs", in *ACM CoNEXT*, 2013.

[7] P. Romirer-Maierhofer, M. Schiavone, A. D'Alconzo, "Device-specific Traffic Characterization for Root Cause Analysis in Cellular Networks", in *TMA*, 2015.

[8] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, H. Zhang, "An Empirical Evaluation of Entropy-based Traffic Anomaly Detection", in *ACM IMC*, 2008.

[9] A. D'Alconzo, A. Coluccia, P. Romirer-Maierhofer, "Distribution-based Anomaly Detection in 3G Mobile Networks: from Theory to Practice", *International Journal of Network Management*, vol. 20, John Wiley & Sons, 2010.

[10] A. Lakhina, M. Crovella, C. Diot, "Mining Anomalies using Traffic Feature Distributions", in *ACM SIGCOMM*, 2005.

[11] H. Ringberg, M. Roughan, J. Rexford, "The need for simulation in evaluating anomaly detectors", in *ACM SIGCOMM Computer Communications Review*, vol. 38, 2008.