Computer Networks 000 (2016) 1–12

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Time-activity footprints in IP traffic

Félix Iglesias*, Tanja Zseby

Institute of Telecommunications, TU Wien, Gusshausstraße 25 / E389, 1040 Wien, Austria

ARTICLE INFO

Article history:
Received 30 October 2015
Revised 1 March 2016
Accepted 23 March 2016
Available online xxx

Keywords:
Communication networks
Traffic characterization
Time domain analysis
Cluster analysis

ABSTRACT

This paper studies the temporal behavior of communication flows in the Internet. Characterization of flows by temporal patterns supports traffic classification and filtering for network management and network security in situations where full packet data is not accessible (e.g., obfuscated or encrypted traffic) or cannot be analyzed due to privacy concerns or resource limitations. In this paper we define a time activity feature vector that describes the temporal behavior of flows. Later, we use cluster analysis to capture the most common time activity patterns in real internet traffic using traces from the MAWI dataset. We discovered a set of seven time-activity footprints and show that 95.3% of the analyzed flows can be characterized based on such footprints, which represent different behaviors for the three main protocols (4 in TCP, 1 in ICMP and 2 in UDP). In addition, we found that the majority of the observed flows consisted of short, one-time bursts. An in-depth inspection revealed, besides some DNS traffic, the preponderance of a large number of scanning, probing, DoS attacks and backscatter traffic in the network. Flows transmitting meaningful data became outliers among short, one-time bursts of unwanted traffic.

1. Introduction

Understanding traffic in communication networks is a challenge that matters not only to network administrators, but also to service providers and security system developers. An accurate knowledge about the structures of underlying communication flows enables fast, proactive reactions against attacks and network degradation, preventing substantial costs. Surveys about cyber crime indicate that the derived costs for companies are immense and increase year by year [1].

When exploring Internet traffic, data analysts confront some difficulties inherent to the field:

- *Big data*. The amount of communication traffic generated every second in the world is overwhelming. Any *representative* dataset easily takes dimensions that overload analysis techniques that are suitable in other areas [2].
- *Evolving nature*. New protocols are used, new applications and new ways to deal with old applications continuously proliferate, so that traffic shapes and rates evolve quickly.
- *Encrypted traffic*. The Internet moves inexorably toward encryption. Recent studies predict that, by the end of 2016, more than two-thirds of the U.S. Internet traffic will be encrypted [3]. This prevents access to packet contents as well as traffic features that have been typically applied for analysis.
- *Fast reaction*. Prompt detection of incidents is required in order to minimize the damage caused by network attacks and breakdowns [4].

In this paper we face the mentioned difficulties by observing the temporal behavior of Internet communications, i.e., registering packet occurrences and transmitted data from a source to a destination during a fixed observation interval. The data object is called a *time-activity vector* and represents a communication flow. The approach is fast, lightweight and non-intensive from the perspective of data preprocessing. Later, we analyzed time-activity vectors with clustering algorithms. In the analyzed captures we discovered that 95.3% traffic followed a set of seven clear patterns (or footprints), which identified specific activities within the most common protocols. These phenomena corresponded to ICMP probing, TCP and UDP scanning, DoS, backscatter and DNS resolution. Clustering-based methods were devised to operate during off-line phases, but clustering outcomes were intended to become filters during real-time monitoring.

Fig. 1 shows a possible scheme for a traffic monitor based on time-activity vectors. This scheme adapts to the *evolving nature* of network traffic as patterns used for classification are periodically updated and refined. *Encrypted traffic* does not make a difference since time-activity vectors only require basic header information that is not encrypted (i.e., source and destination IP

* Corresponding author. Tel.: +4315880138934.
E-mail addresses: felix.iglesias@nt.tuwien.ac.at (F. Iglesias), tanja.zseby@tuwien.ac.at (T. Zseby).
URL: http://www.nt.tuwien.ac.at/about-us/staff/felix-iglesias/ (F. Iglesias), http://www.nt.tuwien.ac.at/about-us/staff/tanja-zseby/ (T. Zseby)

http://dx.doi.org/10.1016/j.comnet.2016.03.012
1389-1286/

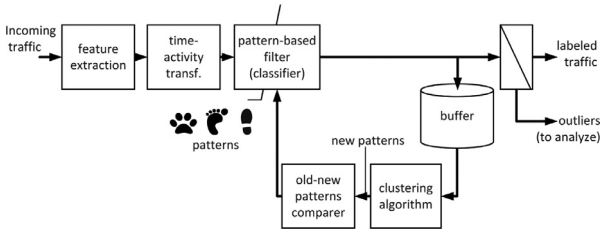

Fig. 1. Scheme of a traffic monitor based on time-activity vectors.

addresses, IP protocol and packet length).[1] A *fast reaction* is guaranteed as the block in the forefront of the process is a simple filter that matches incoming traffic with available patterns; the more costly analysis operations remain in a parallel phase not subject to time constraints. This fact partially alleviates the *big data* problem, which is also addressed by the considerable data reduction involved by the flow representation and the time-activity vector format. In the conducted tests, considering already that captured data has no payload, its size is reduced by about 95.5%, i.e., from 1G to 45M. The bottleneck of this scheme remains in the *feature extractor*, but fast feature extraction is a challenge that any traffic monitor must face.

The benefits of performing traffic analysis by combining clustering and time-activity vectors are:

- *Early filters*. Time-activity footprints are useful filters to preclassify network traffic. Therefore, only a minor part must undergo deeper, more costly analysis to detect sophisticated threats. The complex nature of network traffic forces intrusion detection systems to rely on multi-layer and combined techniques to achieve efficacy [5].
- *Knowledge discovery*. Time-activity footprints provide valuable snapshots to understand how conversations between hosts happen and what the main trends are. Such knowledge leads to the enhancement of networks, protocols and applications, granting strategic information to service providers and network operators.
- *Detection of global events*. Clustering time-activity vectors is suitable for detecting massive and large scale events on the Internet, like scanning activities or DoS attacks. Such incidents respond to multiple, repetitive algorithmic operations that are easy to catch by their time imprint.
- *Analysis of human-in-the-loop*. Even though not covered in this paper and moved to future research, the temporal representation of flows was originally devised to detect the human interaction on the Internet based on the analysis of timings within flows.

Analyses were conducted on datasets from the MAWI Working Group Traffic Archive, which publishes Internet captures for research purposes on a daily basis from 2006 (Section 3). As an additional benefit, our initial analysis tests discovered a misconfiguration problem in MAWI detectors that affected their captures from April to September 2015.

2. Related work

The necessity and difficulties related to an efficient and flexible monitoring for network management have been frequently referred to in the specialized literature [6,7]. The study of Internet flow characteristics, distributions and time behavior is a fundamental strategy to understand how networks are used and can be improved. For example, in [8] statistics of flow durations are investigated, revealing useful knowledge to establish optimized thresholds shortcuts between hosts or network flows. Unidirectional flows are classified based on their inter-arrival-time characteristics in [9], whereas in [10] distributions of inter-arrival-times in flows from large-scale networks are studied and contrasted with flow lengths. In [11] "network elephants" are defined after observing that "a very small percentage of the flows carries the largest part of the information". They are a common phenomenon in backbone traffic and are well isolated by looking at temporal flow characteristics. Authors propose to exploit such a property in traffic engineering applications. In this respect, diverse research groups and projects carry out a continuous measurement of wide area traffic to facilitate the optimization of networking equipment and explain the impact of new protocols [12–14].

In addition, network anomalies, attacks and misconfigurations sometimes present footprints or patterns that can be tracked just by looking at temporal behaviors and packet flows [15]. Internet worms are explored in [16] from this perspective. In [17], a Fourier-based method is applied to detect anomalies after representing traffic by means of graph wavelets that capture the spatial and temporal behavior of Internet flows.

Cluster analysis, as a knowledge discovery method, is an appropriate way to extract and abstract common structures in network traffic data (what is normal) and, by extension, anomalies (what is not normal). This issue is widely discussed in [18], where IP sources are classified by clustering tools and the traffic is represented by means of aggregated features. In [19] behavior profiles of Internet backbone traffic are obtained by using clustering. Source and destination IP addresses as well as source and destination ports are taken to construct a four-dimensional input space where traffic is mapped. Clustering for anomaly detection is also utilized in [20]. In this case the creation of the input space involves some transformations based on entropy and a multiway subspace method. Beyond the specific type of analyzed traffic and the feature level of depth, what usually differs in related works is how network traffic is represented and how clustering techniques are specifically applied.

In our proposal we cluster an input space where flows are drawn as vectors that collect the temporal behavior of flows. In our analysis we do not aim to capture anomalies but footprints of the main phenomena occurring on the Internet datasets. The footprints can then be used to filter traffic and detect deviations from common flow characteristics.

3. Data and features

The analyzed IP traffic traces are measurements of the WIDE backbone, which are available at the MAWI Working Group Traffic Archive.[2] From 2006 on, this archive collects daily sample traces at the transit link of WIDE (150 Mbps) to the upstream ISP. Repositories were firstly introduced in [13,14]. The reasons to choose the MAWI data set for our research are:

- The MAWI dataset is publicly available. It allows further replication of all experiments.
- The MAWI dataset is daily updated. We selected capture files from 2015 in order to reflect most recent trends in the shapes of IP traffic.

[1] Source and destination ports can be used for a more fine granular flow definition. Ports are available with transport layer encryption (TLS), but may be encrypted if network layer encryption (IPsec) is used. In our analysis we extract port numbers from the data for subsequent verification, but they are not required for obtaining the footprints.

[2] http://mawi.wide.ad.jp/mawi/.

Please cite this article as: F. Iglesias, T. Zseby, Time-activity footprints in IP traffic, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.012

- MAWI traces are parts of *a Day in the Life of the Internet* project (DITL) [21], an initiative by multiple organizations for coordinated large scale data capturing throughout the Internet.
- The selected public MAWI captures account for a short time (900 s) but they contain a considerable amount of packets transmitted between many different hosts (between 50 and 200 million packets every 900 s, i.e., between 1 GB and 3 GB of compressed *pcap* traces per file). Such figures entail a significant amount of data about representative Internet traffic.
- MAWI traces reflect snapshots of traffic every day. Thus it is possible to obtain insight into characteristics that prevail or change over time.

We looked at data from January to July 2015. For the in-depth clustering analysis we selected the following MAWI files:

- Id: 201501011400 (*d1* dataset)
  Thu Jan 1 2015, From 14:00:00 to 14:15:00.[3]
- Id: 201504151400 (*d2* dataset)
  Wed Apr 15 2015, From 14:00:00 to 14:15:00.[4]
- Id: 201507311400 (*d3* dataset)
  Fri Jul 31 2015, From 14:00:00 to 14:15:00.[5]

3.1. Data preprocessing

The preparation of the data for the knowledge extraction involved some sequential steps. These steps performed the transformation of *pcap* captures into *time-activity feature vectors*. They are:

1. Extraction of ordered *tuples*. The original *pcap* traces were preprocessed and expressed in a tuple format just containing the following information per packet: timestamp, source and destination addresses, source and destination ports (or *type* and *code* for ICMP packets), protocol, TCP flags, and packet and header lengths.
2. Capturing flows in 60 s activity time-series.
   Relying on timestamps, tuples were swept and the activity of flows was captured in a fixed 60 s time window. We define a *flow* based on the IPFIX definition in RFC7011 [22]; hence, a flow is the unidirectional data stream between a sending host A and a receiving host B, i.e., "A > B" ("B > A" would be a different flow). Although IPFIX also allows bidirectional flows [23], we decided to work with unidirectional flows as it requires less analysis steps and no state keeping. Furthermore, the generation of the time activity vectors involves packet arrival times and therefore can anyway not be directly generated from IPFIX flow data.
   An example of the time series of a flow is displayed in Fig. 2. Flows whose duration was shorter than 60 s showed a 0-tail in the right side of the time series; flows with a duration longer than 60 s were split and considered as separate 60 s flows for the analysis. Flows which could not be observed in a time frame of at least 60 s were directly discarded (i.e., residual heads and tails of the analyzed files).
3. Summarizing into time-activity feature vectors. Some characteristic features were extracted from the activity time-series. The definition and format of such features are explained in detail in the next subsection, Section 3.2.

3.2. Time-activity vector

Traffic flows were finally represented by *time-activity feature vectors* (or just *time-activity vectors* for brevity), which were ultimately the objects of the knowledge discovery processes. Time-activity vectors were formed by the fields listed in Listing 1.

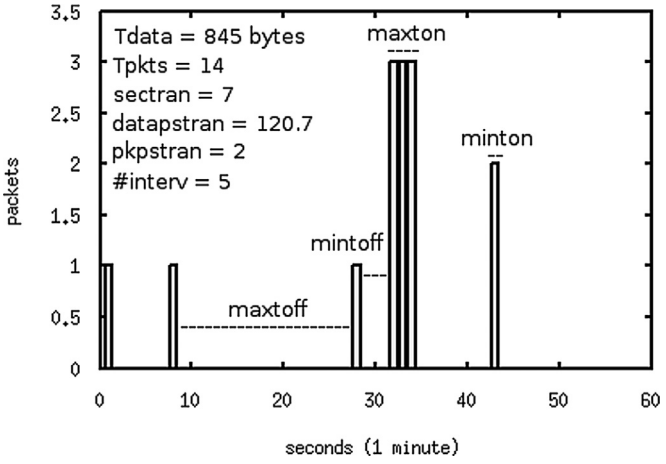

Fig. 2. Time series example of an IP flow time-activity vector.

```
{timestamp , flowID , protocol , ports ,
Tdata , Tpkts , sectran , datapstran , pkpstran ,
maxton , minton , maxtoff , mintoff , interv}
```

Listing 1. Time-activity vector format.

Timestamp, flowID are used as flow keys to identify a flow. *Protocol* and *ports* are also collected to facilitate a deeper analysis. A flow in our definition always lasts 60 s. *timestamp* stores the time (resolution of seconds) when the flow started. *FlowID* retains the source and destination addresses of the flow hosts with a ' > ' to mark flow direction, e.g., "A > B". *Protocol* contains the protocol number characteristic of the flow. *Ports* saves either the UDP and TCP ports with the format: *source-port:destination-port* or, in the case of ICMP, the pair: *type:code*. For other protocols that do not use port numbers, the value of *ports* is set to "0:0". In the cases where the flow uses various protocols or ports, the respective fields take the '–1' value.

The remaining fields are:

- *Tdata*: total amount of data transmitted (bytes).[6]
- *Tpkts*: total amount of packets transmitted.
- *sectran*: number of seconds when the flow was *active*.
- *datapstran*: average data per active-second transmitted (bytes).
- *pkpstran*: average packets per active-second transmitted.
- *maxton*: maximum amount of consecutive seconds that the flow showed activity.
- *minton*: minimum amount of consecutive seconds that the flow showed activity.
- *maxtoff*: maximum amount of consecutive seconds that the flow did not show activity.
- *mintoff*: minimum amount of consecutive seconds that the flow did not show activity.
- *interv*: number of activity intervals.

For analysing the time-activity, the sampling granularity within the 60 s flow was 1 s.

Time-activity feature vectors were formed by transforming flow-activity time series. As an example, given a time series like the one in Fig. 2, which accounts for a TCP flow between host 211.74.5.25 (port 1234) to host 146.34.9.14 (port 80), starting at

[3] http://mawi.wide.ad.jp/mawi/samplepoint-F/2015/201501011400.html.
[4] http://mawi.wide.ad.jp/mawi/samplepoint-F/2015/201504151400.html.
[5] http://mawi.wide.ad.jp/mawi/samplepoint-F/2015/201507311400.html.

[6] *Tdata* is calculated as the length of the IP datagram minus the length of the IP header and the length of the TCP (or the UDP) header. ICMP and other protocols headers are considered as payload.

```
{1439300313,211.74.5.25>146.34.9.14,6,1234:80,
845,14,7,120.7,2,3,1,19,3,5}
```

Listing 2. Time-activity vector example.

13:38:33 GMT on Tue 11 Aug 2015; the corresponding time-activity vector would remain as displayed in Listing 2:

4. MAWI dataset overview

We here describe some general characteristics of the traffic collected in the MAWI datasets.

4.1. Data and packet rates

The MAWI webpage contains some descriptive information about the traffic captures. From 1 January to 31 July 2015 we used the daily rates, reported at the MAWI page, to study the evolution of traffic according to different protocols. The number of packets and bytes of every 900 s sample per day are shown in Figs. 3 and 4. Beyond confirming the preponderance of TCP, UDP and ICMP traffic, from the figures we observed some initial traits about the nature of IP traffic in MAWI captures:

- Most of the data traveling the network belongs to TCP transmissions. TCP is the dominant protocol on the Internet, so this observation was expected.
- TCP traffic packet and data rates follow a weekly pattern, from which low peaks usually coincide with saturdays and sundays. Higher peaks during work days are also typical in Internet traffic.
- UDP traffic showed a chaotic behavior, with occasional high peaks of packet rates (usually transporting low amounts of data).
- The ICMP number of bytes was low and very stable. ICMP time series related to number of packets showed a flat shape with two well differentiated steps.[7]
- There was a considerable increment of global packet and data rates as of end March (data) and mid May 2015 (packets).[8]

4.2. Pre-analysis of flows

The amount of observed ICMP packets in the MAWI datasets was considerable. In this respects, MAWI data publishers warn about the unusually large amount of ICMP traffic in the traces,[9] mostly caused by the USC ANT project [24,25]. The USC ANT project uses ICMP to probe the entire IPv4 space and is constant in a high activity rate since March 27, 2013.

When analyzing the situation from the perspective of time-activity flows, the predominance of ICMP traffic is unquestionable (Table 1 and Fig. 5). The provided figures show that most of the flows consisted of short, low-data ICMP packets.

5. Knowledge extraction methodology

Beyond digging into the superficial characteristics of IP flows, our exploration of the MAWI datasets aimed to discover time-activity patterns able to identify big portions of the IP space without checking packet contents and only few parts of the packet headers. Once preprocessing steps were finished, i.e., all captured data was transformed into time-activity feature vectors, the knowledge extraction analysis was carried out according to the following steps (Fig. 6 displays the corresponding scheme):

1. Subset sampling.
   Given the huge amount of data to deal with, for every *global set* (i.e., *d1, d2* and *d3*) we took several subsets of about 5% to 10% of the samples by means of a random permutation algorithm – we call them *analysis subsets*. We used such subsets to find statistical properties, to adjust analysis parameters and, after separating into protocols, to discover significant clusters. We assumed that the statistical power of the subsets was enough to guarantee their representativeness. The validity of this assumption was checked later when applying the derived methods and filters to the whole data.
2. Descriptive analysis of features.
   The subsets were submitted to statistical analysis to evaluate their boundaries and superficial shapes. Analysis outcomes disclosed that all features showed skewed, unbalanced distributions. This fact was specially significant for features with a high dynamic range, i.e., the ones related to total or averaged amount of data or packets (*Tdata, Tpkts, datapstran, pkpstran* — see Section 3.2).
3. Correlation analysis.
   Correlation analysis provide outcomes that explain dependencies in feature variations, therefore helping to understand the nature of the explored datasets. Such information allows to refine the selection of clustering parameters and methods, as well as to suggest redundancies for the subsequent extraction of the most relevant features, described in Section 10. In this respect, all features turned out to be positively or negatively correlated with at least one other feature (coefficient absolute values above 0.7). The couples *Tdata–datapstran* and *maxton–minton* showed the strongest linear correlation. High correlation coefficients do not necessarily mean that some features are dispensable for the traffic representation; little differences can be determining for the final classification. Therefore assessments about feature selection should not be done prior to the clustering analysis (unless the number of features were overwhelming). They must be performed after post-processing steps (Section 10), since clustering outcomes are necessary to validate the feature selection.
4. Logarithmic transformation of features with highly skewed distributions.
   Features with unbalanced distributions within wide dynamic ranges can hamper the efficacy of subsequent cluster analysis. In such cases, capturing orders of magnitude (tens, hundreds, thousands) instead of absolute values can be advisable, still providing meaningful information while presenting a more manageable input space to the analysis algorithms [26]. Following such reasoning, *Tdata, Tpkts, datapstran, pkpstran* were transformed according to Eq. 1. An example of dynamic ranges and feature distributions is shown in Fig. 7.

$$y = 1 + \log_{10} x \tag{1}$$

5. Normalization.
   To equalize the importance of features during the analysis, datasets must undergo some kind of normalization. Given the characteristics of the features, we opted for *range* normalization (Eq. 2), i.e., to divide each sample value by feature maximums and minimums, therefore the dynamic range of every feature remains enclosed between [0...1].

$$y_i = \frac{x_i - \min(X_i)}{\max(X_i) - \min(X_i)} \tag{2}$$

[7] Some misconfiguration problems affected MAWI monitors from end May 2015 to early September 2015, generating a considerable amount of duplicated packets. We discovered this issue with the application of clustering-based analysis (see Section 6). Figs. 3 and 4 are therefore affected and can lead to wrong interpretations.

[8] See footnote [7].

[9] http://mawi.wide.ad.jp/mawi/ .

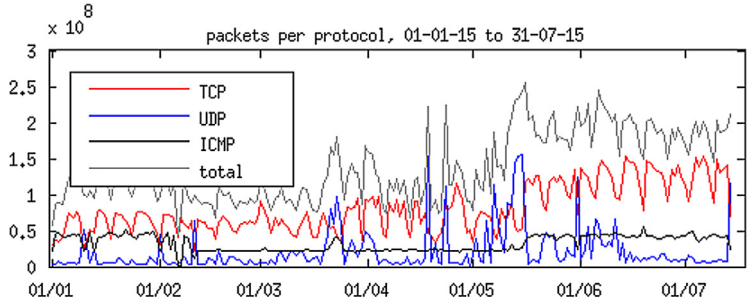

Fig. 3. Daily amount of IPv4 packets from 14:00 to 14:15 split into protocol-type, from 1 Jan to 31 Jul 2015.

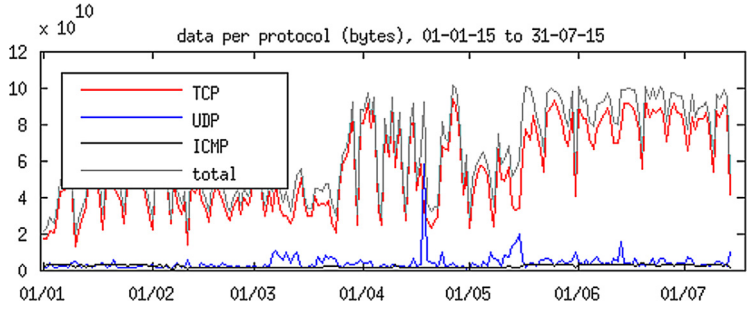

Fig. 4. Daily amount of IPv4 data (bytes) from 14:00 to 14:15 split into protocol-type, from 1 Jan to 31 Jul 2015.

Table 1
Distribution of IP flows in protocols.[a]

	01-01-2015	15-04-2015	31-07-2015
TCP	1.3 M (07.3%)	1.6 M (08.6%)	4.7M (31.9%)
ICMP	16.4 M (89.4%)	16.1M (88.1%)	9.5 M (64.5%)
UDP	486 K (02.7%)	604 K (03.3%)	473 K (03.2%)
Multi	130 K (00.7%)	10 K (00.1%)	58 K (00.4%)
Others	172 (neglig.)	192 (neglig.)	169 (neglig.)

[a] After removing duplicates (see Section 6) as well as heads and tails – i.e., considering about 780 s of traffic.

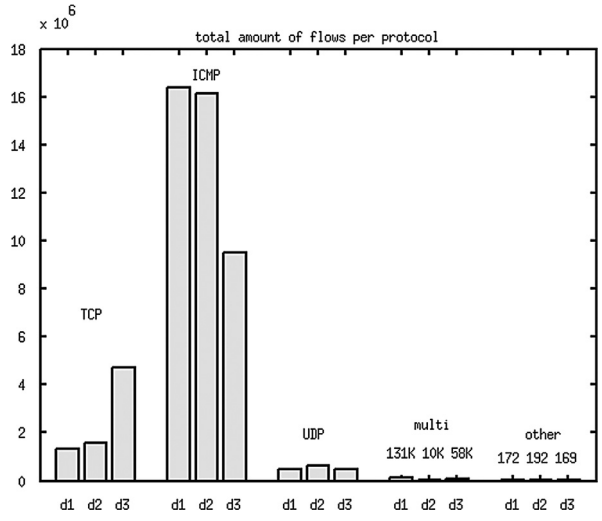

Fig. 5. Number of flows according to protocol type. *d1, d2* and *d3* stand for 01/01/15 dataset, 15/04/15 dataset and 31/07/15 dataset respectively.

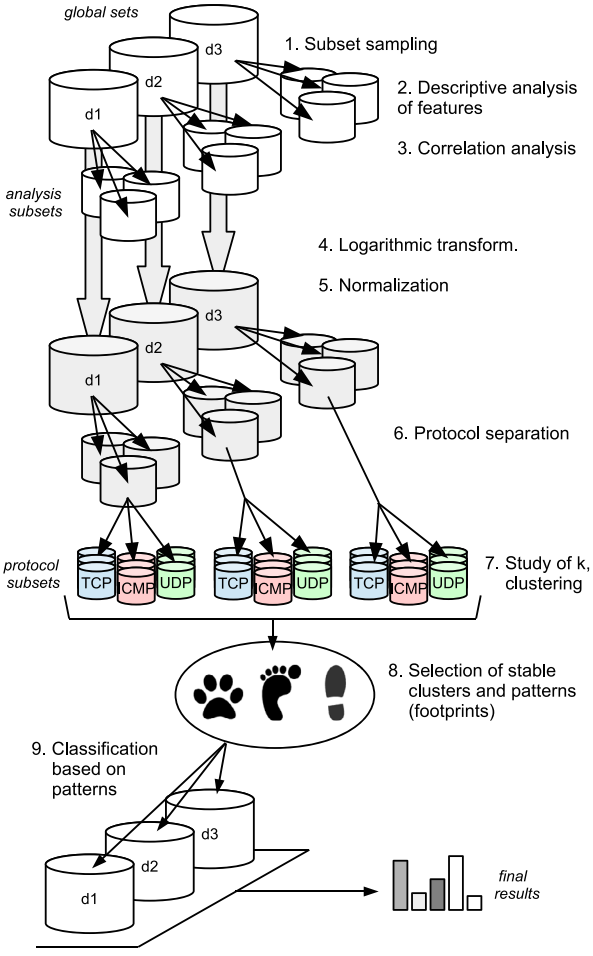

Fig. 6. Conducted steps linked to the respective data subsets.

where i identifies the specific *feature* and X_i stands for the subset with all the possible values taken by feature i.

6. Separation according to protocols.
   The subsets were separated once more into smaller subsets according to the main protocols: TCP, UDP, ICMP – we call them

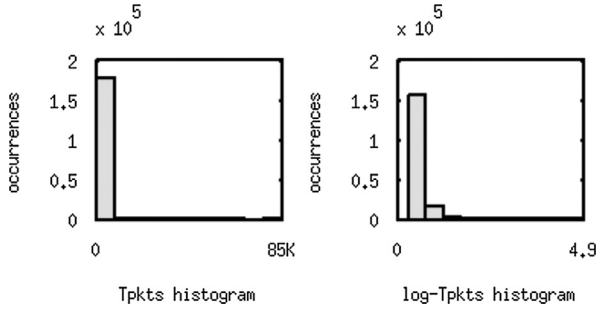

Fig. 7. Histograms of the feature *Tpkts* along the *d1* dataset, before and after logarithmic transformation.

protocol subsets. The purpose was to find the representative clusters and patterns of every main protocol.

7. Study of *k*, number of clusters. Clustering.
   For every protocol subset the suitable number of clusters *k* was explored. As a design parameter, for the model discovery we limited *k* between 2 and 15. The analysis of the most suitable *k* in every case was carried out by weighted votes among diverse cluster validation indices. We applied: *classification entropy, partition index, Xie and Benix's index* [27,28], *clustering gain* [29] as well as an *ad-hoc* validity method based on inter- and intra-cluster distances.
   The selected clustering algorithm was the Fuzzy Gustafson-Kessel clustering method with squared Mahalanobis distance norm [30] and outlier rejection. Fuzzy clustering was chosen to avoid undesired local minima problems that could distort the solution [31], also to deal with a noisy environment and overlapping clusters in a more efficient way than with crisp clustering [32]. The Mahalanobis distance based Gustafson-Kessel approach responds to the suspicion of elongated clusters (due to features with variated statistical behavior) and to the evidence of high correlated data and features, disclosed during the correlation analysis (see current section, list-point 3). This method allows that features with low variance have more importance for the subspace division than features with high variance [33]. Since we looked for discovering clusters and patterns representing most of the population under analysis, removing outliers based on the Median Absolute Deviation (MAD) [34] was appropriate. The removal of outliers enabled the obtaining of cleaner centroids.
8. Selection of big, stable clusters and patterns.
   From all the discovered clusters, we kept the ones that embraced a considerable amount of samples in the analyzed datasets (above 1% of the total flows) and showed acceptable quality coefficients based on inter- and intra-cluster distances.
   For every cluster a pattern was established. A pattern is a special representative sample that accepts some drifts and redefines cluster boundaries. Cluster boundaries are defined according to Eq. 3:

$$patt_{a,i} = centroid_{a,i} \pm 2 \times intra\_dist_{a,i} \quad (3)$$

   where $patt_{a,i}$ identifies the value of feature *i* in pattern $patt_a$, $centroid_a$ is the centroid of cluster *a*, and $intra\_dist_{a,i}$ is the average distance to $centroid_a$ of all samples embraced in cluster *a*.
   In short, a pattern describes a subspace (specifically a hyper-rectangle of *n*-dimensions) within the input space defined by the *n* features of the *time-activity feature vector* format. From the perspective of traffic analysis, a cluster pattern is a traffic footprint.
9. Classification of the whole datasets with the discovered patterns. With the final set of patterns already defined, all samples in the whole *d1, d2* and *d3* datasets were filtered and classified. Results are displayed and discussed in Section 8.

6. Detection of misconfiguration in captures

The outcomes of the first analysis disclosed very similar clusters in datasets *d1* and *d2*, but a strong alteration in major flow shapes and rates of the *d3* dataset. In short, the *d3* dataset contained clusters that accounted for a significant amount of traffic, but were nonexistent in *d1* and *d2*, and their shapes showed doubled values in *Tdata, Tpkts, datapstran* and *pkpstran* when compared to other clusters. The manual inspection of captures in *d3* exposed that such clusters represented flows with duplications – i.e., a second identical packet – captured some micro- or milliseconds after the first packet.

The issue was communicated to MAWI on early September 2015. After a few days, MAWI experts confirmed a misconfiguration problem affecting one of their routers. The misconfiguration caused the re-injection of a significant portion of traffic and affected published captures from 28 May 2015 to 3 September 2015.

Published datasets have been cleaned by MAWI technicians after the discovery of this problem. We repeated the experiments with cleaned datasets, therefore results displayed in Table 1 and Sections 8, 9 and 10 have been obtained after the removal of duplicates.

7. Analysis times

The goal of the research was the exploration and knowledge extraction of the captured traffic, and not optimizing the computational effort of the analysis. Table 3 provides some guiding figures to offer a general impression of the preprocessing and analysis costs with preliminary, non-time-optimized programming. All calculations have been performed with scripts built over the following softwares and programming languages: TShark[35], Python[36], Perl[37] and MATLAB[38].

Figures in Table 3 correspond to the preprocessing of the d3 dataset and the analysis of d1, d2 and d3 together. Note that d3 is the biggest dataset; it consists of 213758683 packets (almost 214 millions), 11.2 GB of captured header data with no payload.

8. IP flow footprints

The discovered clusters were stable throughout every dataset, yet their relative size varied depending on the specific dataset under analysis. The selection of the appropriate number of clusters *k* is usually an arguable parameter, submitted to the peculiarities of the clustering methodology. We applied a weighted evaluation of diverse validity methods to fix *k* (Section 5, point 7). In any case, in our analysis an accurate selection of the initial *k* is not a crucial decision as, in a posterior step, we remove minor, less representative clusters and focus only on the big, clear partitions (Section 5, point 8).

The patterns of the principal discovered clusters are displayed in Table 2. We denote clusters found in TCP traffic with T1, T2, T3 and T4; clusters from UDP flows with U1, U2; and the cluster representing most ICMP traffic with I1. They embraced the following amount of flows:

T1: 0.6% T2: 5.0% T3: 5.6% T4: 0.3%
I1: 79.9% U1: 0.4% U2: 3.5% out.: 4.7%

The total analyzed data accounted for 51.4M flows. 48.9M out of 51.4M flows (95.3%) matched patterns in Table 2. In spite of the differences, patterns show some characteristics that are common in the analyzed traffic. Moving apart small clusters T1 and T4, the

Please cite this article as: F. Iglesias, T. Zseby, Time-activity footprints in IP traffic, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.012

Table 2
Pattern cores of the main discovered clusters: central value (in brackets ranges covered by the first standard deviation).

	T1	T2	T3	T4	I1	U1	U2
Tdata (B)	0	896	0	0	12	242 (42,1384)	56 (36,86)
Tpkts	2 (2,3)	1	1	4 (3,6)	1	2 (2,6)	1
Sectran (s)	2 (2,3)	1	1	3 (2,4)	1	1 (1,2)	1
Datapstran (B)	0	896	0	0	12	229 (44,1169)	56 (36,86)
Pkpstran	1	1	1	1 (1,2)	1	2 (2,5)	1
Maxton (s)	1	1	1	1 (1,2)	1	1 (1,2)	1
Minton (s)	1	1	1	1	1	1 (1,2)	1
Maxtoff (s)	55 (53,57)	59	59	48 (43,52)	59	59 (57,59)	59
Mintoff (s)	2 (1,3)	59	59	3 (1,7)	59	59 (58,59)	59
#Interv	2 (2,3)	1	1	3	1	1	1

Table 3
Analysis times.

Preprocessing[a] (d3 dataset)		
-Remove duplicates, split in smaller sets	*Editcap*	06 m 40 s
-Extracting header features	*TShark, Python*	15 h 06 m 44 s
-Parsing, preprocessing	*Perl*	17 m 12 s
-Extracting time series	*Perl*	28 m 27 s
-Extracting time-activity vectors	*Perl*	06 m 20 s
-Final parsing, preprocessing	*Perl*	01 m 09s
Analysis[b] (d1, d2 and d3 datasets)		
-Importing data, subset sampling, descriptive and corr. analysis, log. transformation	*MATLAB*	04 m 26 s
-Importing data, normalization, clustering, outlier removal, obtaining indices, validity comparisons	*MATLAB*	28 m 22 s
-Importing data, classification based on patterns, results aggregation	*MATLAB*	3 h 16 m 34 s

[a] Machine: 64-bit, Intel Core i7-4770T CPU @ 2.50GHz x 8, 16GB RAM, Ubuntu 14.04 LTS.
[b] Machine: 64-bit, Intel Processor 5Y10 CPU @ 0.80GHz 4, 8GB RAM, Ubuntu 14.04 LTS.

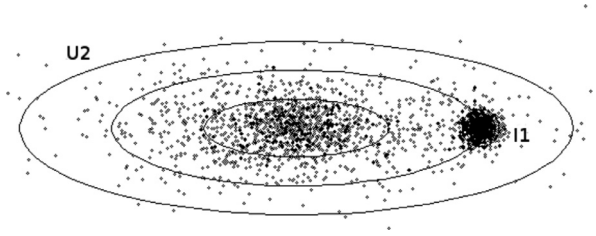

Fig. 8. Idealized two-dimensional representation of the solution space to illustrate the pair I1-U2 case. Concentric circles mark *global* distance radius according to standard deviation multiples.

most noticeable common trait of the remaining clustered traffic (94.2%) is that flows between hosts were generally short, one-time bursts of one to few consecutive packets transmitted immediately.

Although cluster cores are well differentiated, some overlapping among clusters boundaries appears. For instance, I1 is an extremely dense cluster inside the confines of a less dense cluster U2. Fig. 8 projects an idealized two-dimensional representation of the solution space in order to illustrate this case. Something similar happens with T3 and U2, or T2 and U1. This is the reason why a considerable amount of UDP traffic matched I1 during the analysis and, in a minor proportion, also matched T2 and T3. To check these aspects, Table 4 shows how TCP flows distributed into the discovered clusters, analogously Table 5 for ICMP flows, and Table 6 for UDP.

Deep down what such overlapping expresses is that some flows from different protocols behaved in a similar way when considering only time-activity footprints. Up to a certain degree this is expected and is not a critical issue if the underlying phenomena behind is understood. For example, if we can identify which type of UDP traffic falls into I1. We analyze such aspects

Table 4
Distribution of TCP flows in clusters.

	01-01-2015	15-04-2015	31-07-2015
T1	7.0%	4.9%	2.9%
T2	0.3%	0.8%	53.0%
T3	57.7%	71.2%	20%
T4	6.6%	1.0%	1.0%
I1	0.1%	0.1%	0.1%
U1	2.3%	2.9%	0.7%
U2	0.1%	0.3%	0.1%
Out	26.0%	18.7%	22.2%

Table 5
Distribution of ICMP flows in clusters.

	01-01-2015	15-04-2015	31-07-2015
T1	0.0%	0.0%	0.0%
T2	0.0%	0.1%	0.0%
T3	0.0%	0.0%	0.0%
T4	0.0%	0.0%	0.0%
I1	97.9%	97.8%	95.3%
U1	0.1%	0.1%	0.0%
U2	1.5%	1.7%	3.5%
Out	0.5%	0.5%	1.1%

Table 6
Distribution of UDP flows in clusters.

	01-01-2015	15-04-2015	31-07-2015
T1	0.0%	0.0%	0.0%
T2	2.4%	1.4%	4.7%
T3	0.2%	7.2%	0.0%
T4	0.0%	0.0%	0.0%
I1	3.9%	12.9%	7.3%
U1	5.9%	5.7%	5.2%
U2	58.2%	59.5%	63.7%
Out	29.5%	13.3%	19.1%

Table 7
Distribution of data and packets according to clusters.

Cluster	Flows	Bytes	Pkts	Bytes/flow	Pkts/flow
T1	0.6%	0.0%	0.2%	0.0	2.1
T2	5.0%	2.1%	1.0%	1.0 K	1.1
T3	5.6%	0.0%	1.1%	0.1	1.0
T4	0.3%	0.0%	0.2%	0.0	3.4
I1	79.9%	0.4%	15.3%	12.0	1.0
U1	0.4%	0.1%	0.4%	0.4 K	4.7
U2	3.5%	0.1%	0.7%	71.7	1.0
OutTCP	3.2%	91.1%	48.9%	61.7 K	79.1
outICMP	0.5%	0.1%	1.2%	0.5 K	12.0
outUDP	0.6%	4.3%	28.7%	15.1 K	0.2 K
outMulti	0.4%	0.4%	0.9%	2.3 K	12.0
outOther	0.0%	1.4%	1.3%	3.1M	6.9 K

out{TCP,ICMP,UDP}: TCP, ICMP and UDP flows clustered as outliers. outMulti: outlier flows that use multiple protocols. outOther: outlier flows that are not TCP, ICMP, UDP or multi-protocol.

later in this section when describing the traffic represented by the footprints. In any case, it is noteworthy that, from a global perspective, Tables 4, 5 and 6 display how the analyzed TCP, ICMP and UDP traffic flows were distributed mostly fitting the patterns corresponding to their specific protocol. Hence, in the analyzed captures, time-activity vector patterns can be used to identify the protocol of a considerable part of IP traffic. These results can be better assessed with some figures:

- 93.2% of all TCP, ICMP and UDP flows fell in clusters corresponding to their protocol type (76.5%, 97.3% and 64.7% respectively).
- 4.4% of TCP, ICMP and UDP flows were labeled as outliers (21.8%, 0.6% and 20.5% respectively).
- 2.4% of TCP, ICMP and UDP flows fell in a non-corresponding protocol type cluster (1.7%, 2.1% and 14.9% respectively).
- 96.5% of flows using more than one protocol type (about two hundred thousand) as well as flows using protocols different to TCP, UDP and ICMP (slightly above five hundred) were labeled as outliers.

Another remarkable fact observable in Table 4 is an evolution of TCP cluster distributions that occurred in *d3*. In the following section we give an explanation to this fact.

8.1. TCP patterns

Pattern T1: The manual inspection of flows matching T1 disclosed TCP connection attempts mainly to ports 445 and 23. The standard case behind this footprint was a source trying to connect to a destination with a SYN packet. After getting an ICMP Destination Unreachable, a TCP RST or no response from the destination, the source tried to connect a second time (sometimes even a third time).

Such a situation was unique or exceptional to sources with flows clustered under T1, hence in principle T1 is not necessarily an indication of illegitimate behavior. However, part of the flows generated by a source that was conducting a massive scan to port 23 in *d2* fell into this group. In such a case, the used scanner occasionally sent a second SYN packet (a different packet, not a duplication) to just some destinations after some seconds. This behavior is not usual in the observed TCP scanning, it might be caused by a peculiarity or defect in the specific exploration algorithm.

Pattern T2: This cluster was caused by an isolated address – we refer to it as *add0* – that concentrated 98.7% of all TCP flows in T2. T2 flows showed one SYN packet transmitting about 896 bytes of data (excluding TCP and IP headers) to *add0*, specifically to the TCP destination port 80. The SYN flag was sometimes combined with less habitual flags, e.g., ECN, CWR, NS, etc. *add0* did not answer or start any communication attempt in the analyzed traffic. We hypothesize that T2 captured a new generation of intense DoS attack known as Tsunami SYN flood [39].

The remaining, spurious TCP flows clustered by T2 not identifiable as belonging to the Tsunami attack consisted mostly in very short, legitimate bursts delivered by web servers (source ports 80 and 443); to a minor degree, also going to web servers (destination ports 80 and 443). UDP flows falling into T2 corresponded to DNS resolution – mainly answers – that matched the characteristic payload size of the footprint. Such web server TCP activities and DNS resolution captured by T2 were often flows in the boundaries of the T2 cluster, exhibiting drifts essentially in the number of transmitted packets (more than one).

Pattern T3: The manual inspection of flows matching T3 revealed a clear profile of scanning behavior: no data, one TCP packet with the SYN flag, few sources and many different destinations from selected address ranges. The preferred ports to scan varied significantly depending on the analyzed file *d1, d2* or *d3*. This is due to the fact that most flows came from a few number of very active sources, which were aiming at different ports in every analyzed period. In any case, ports 443, 8888, 445, 22, 23, 1080, 11211 and 3389 seemed to be classic objectives for scanning since they appeared always within the top 25 most scanned ports for the three analyzed datasets.

This pattern also embraced answers to the scanning activity, not only from protected hosts responding with RST or RST-ACK, but also in a few cases from vulnerable hosts accepting the connection with a SYN-ACK response (but never receiving a corresponding ACK from the source). The amount of flows showing scanning reactions were much lower than the scanning activity itself, presumably due to the fact that scanners point to random addresses that can be nonexistent as well as many scanned hosts incorporate dropping policies against undesired connection attempts or directly send ICMP messages to inform about the rejection.

UDP flows falling in T3 corresponded to UDP scans with no payload. In the visual inspection, most of such scan pointed to UDP port 1900 (also UDP port 19 was aimed at in a reduced scale).

Pattern T4: The manual inspection of TCP flows in T4 revealed diverse cases:

- Servers rejecting retransmitted connection attempts with RST-SYN packets. Servers could not satisfy the connection requests for some reason.
- Retransmitted SYNs from clients (mainly to ports 80, 443 and 23). In this case, servers were overloaded, suffering a DoS attack or they just rejected connection attempts for some reason.
- Servers sending several late SYN-ACK packets (mainly from ports 80 and 443). The tracking of such hosts disclosed clear cases of servers suffering DoS attacks, already overloaded or quickly becoming overloaded. SYN-ACK packets were sent to presumably spoofed sources while waiting for never-coming ACKs.

8.2. ICMP patterns

Pattern I1: The manual inspection of flows matching I1 showed that most of such traffic corresponded to Echo requests (pings) from at least four specific IP addresses as well as Echo replies going to these same addresses (we refer to them as *add1, add2, add3* and *add4*). Only 0.9% of flows that matched pattern I1 did not involve *add1, add2, add3* or *add4*. We hypothesize that such addresses belonged to the ANT project probing mentioned in Section 4.2. Therefore, ICMP flows from *add1, add2, add3* and *add4* matching I1 covered respectively 79.1% of the total analyzed

flows. In other words, approximately 79 out of 100 flows in MAWI datasets were related to the ANT project probing activity.

I1 also included other ICMP Echo request and reply flows not linked to the ANT probing. In addition, TCP flows that very occasionally fell in I1 corresponded to tails of longer TCP connections that were cut by the 60 s time-window defined for the time-activity vector format.

As for the considerable amount of UDP flows falling in I1, the visual inspection revealed the following situations:

- Sources performing UDP scans (in the observed files, mainly to ports 161, 9987, 5351 and 53).
- Isolated communication attempts with inactive or unresponsive UDP services.
- At a negligible rate, residuals of legitimate UDP communications (tails of longer UDP conversations).

8.3. UDP patterns

Pattern U1: 76.8% UDP flows included in U1 mainly belonged to legitimate DNS resolution (UDP source and destination port 53). At a lower rate (2.3%), this footprint also embraced some flows of legitimate bit-torrent communications. Such behavior consisting of instantaneous, isolated bursts of a few UDP packets seemed to be quite common in both DNS and bittorrent services. Also some UDP scans were observed falling into this group; for example, scans to UDP port 123 that were truncated in two packets by network analysis tools due to the long packet size, or scans actually sending two consecutive packets to port 161. The remaining residual flows belonged to normal UDP communications that matched the footprint.

Finally, the scarce number of TCP flows falling in U1 corresponded to legal, short TCP connections whose data and packet rates matched U1.

Pattern U2: U2 clustered many UDP scanning activities, where ports 53 (DNS), 161 (SNMP), 123 (NTP) and 1900 (SSDP, UPnP) turned out to be the most targeted ones in the analyzed files. Activities related to these ports accounted for 84.2% of the flows. Again, beyond scanning activities, this footprint is also characteristic of flows with non-suspicious DNS queries and answers (approximately one third of the traffic related to port 53 was normal DNS traffic), as well as some flows of residual bittorrent communications.

On the other hand, a considerable amount of ICMP traffic fell in U2. The visual inspection of such flows disclosed a dominant presence of ICMP Destination Unreachable messages, mainly from hosts that were being scanned (about 70%). It also included ICMP Time Exceeded messages, Echo requests and Echo replies. The difference between I1 and U2 relies on the payload size, that in the case of Destination Unreachable messages is bigger as it contains part of the original, not successfully delivered packet. Some Echo request and replies fell in U2 because this type of ICMP packets admit the addition of variable contents in the payload.

8.4. Other protocols or multi-protocol flows

The number of flows using other protocols or combinations of protocols were negligible if compared with the rest of the traffic. Out of 44.7 M flows, 198 K were flows with packets from diverse protocols and less than 1 K came from protocols different to TCP, ICMP or UDP. In any case, 96.5% of such flows did not match any pattern and were classified as outliers.

9. Minor clusters and outliers

After filtering all flows belonging to the discovered seven pattern set, the remaining flows that, either belonged to minor clusters or were directly identified as outliers, accounted for the 4.7% — i.e., 2.4 M out of 51.4 M flows. Outliers were mainly TCP flows (68.0%), followed by UDP and ICMP flows (13.1% and 11.0% respectively). The remaining 7.9% corresponded to multi-protocol flows or flows using other protocols different from TCP, ICMP or UDP. Fig. 9 illustrates such figures with a chart.

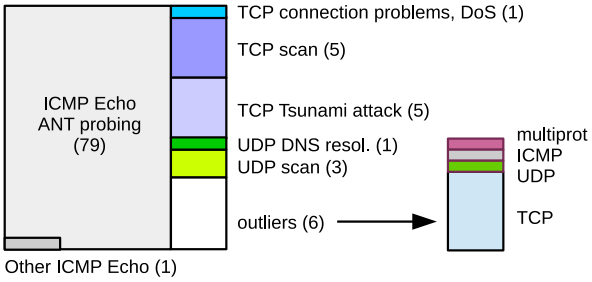

Fig. 9. Distribution of 100 random flows from the analyzed MAWI datasets after stratified sampling.

In the *outlier subset* we found the actual delivery of real and meaningful data over the Internet. For instance, the duration of active intervals was significantly longer in the outliers and distributed along the 60 s time window. In high contrast with the clustered flows, which mostly occurred in an isolated interval (Table 2), 87.1% of outlier flows showed two or more intervals in the activity profile, i.e., source hosts sent their packets in, at least, two bursts separated by one second of no activity. Multi-interval flows were 75.5% TCP, 10.3% ICMP and 6.8% UDP.

To check the number of intervals or their duration is revealing in order to disclose where the exchange of actual information is located. Directly evaluating data rates can be misleading since, for example, we have seen how Tsunami attack flows massively sent meaningless, heavy payloads just to overload the attacked host and bypass security barriers. Also some UDP scans carried data or ICMP packets stored data about the packet that originated the ICMP message. This fact can distort slightly the perception of "elephants" as envisioned in [11], yet their existence is obvious when packets and data histograms are checked (Section 5, point 4; also Fig. 7). Looking for long-active-time flows is also a good evaluator to discover where the actual information transfer took place. For instance, flows with a *sectran* $\geq$ 5 s (all of them classified as outliers) accounted for 0.13% of the total flows, being 70.3% TCP, 5.2% ICMP and 19.1% UDP.

Although classified as outliers, we found that most ICMP multi-interval flows also pertained to the ANT probing activity, but in this case the strategy consisted in sending a second Echo request (a new packet, no duplicate) to every scanned destination 2 or 3 s after the first. This otherwise well-defined group was not included within the main patterns due to its low representativity compared to I1, but accounts for about 95% of the ICMP outliers displayed in Table 5.

Finally, Table 5 shows a relationship between flows, data (bytes) and packets, as well as data and packets per flow. It is clearly visible how most of the data exchange belongs to the outliers, mostly TCP outliers.

In short, TCP traffic showed to be the richest and most variable type of traffic in terms of time-activity behavior, containing most of the multi-interval and long-active flows and being the common protocol within the outlier group (Table 7). Future research must focus on such traffic and on the identification of TCP and UDP services by means of time-activity footprints once the bulk of probing and scanning activities as well as other massive flow-types have been filtered. Time-activity vectors were also conceived to capture flows where the human interaction appears, possible to identify thanks to the time-between-packets or time-between-intervals response, which is supposed to differ

Please cite this article as: F. Iglesias, T. Zseby, Time-activity footprints in IP traffic, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.012

Table 8
Precision and recall indices. Using 1ten features (left) and four features (right).

	Precision	Recall	Precision	Recall	
T1	99.18%	98.37%	99.19%	99.19%	T1
T2	99.65%	99.74%	98.96%	99.57%	T2
T3	99.91%	100.00%	100.00%	100.00%	T3
T4	96.61%	98.28%	93.22%	94.83%	T4
I1	100.00%	99.99%	100.00%	100.00%	I1
U1	82.76%	84.00%	82.96%	84.00%	U1
U2	84.69%	83.84%	86.23%	83.84%	U2
Out.	99.68%	99.37%	99.37%	99.16%	Out.

if compared with scheduled or algorithmic machine-to-machine operations.

10. Most relevant features

Given the high feature correlation and the similarities observed among the most common footprints, we expected to find irrelevant or redundant features in the time-activity vector when applied to classify traffic flows based on the discovered patterns. To check the feature selection, we randomly selected subsets of samples (stratified sampling) from the three datasets linked to their characteristic footprint. The subsets underwent a feature selection filter based on correlation and maximum-relevance minimum-redundancy measurements [40]. Results emphasized:

Tdata, datapstran, maxtoff and *mintoff*

as the most relevant features. Table 8 displays the results obtained from the confusion matrix[10] after applying 1ten-fold cross-validation with a *k-NN* classifier [42] in both: the complete ten-feature subset and the reduced four-feature subset.

Results of the feature selection analysis revealed that traffic flows can be either classified according the discovered patterns or labeled as outliers just by considering only *Tdata, datapstran, maxtoff* and *mintoff*. But the remaining features cannot be discarded for further analysis that aim to discover footprints inside the outliers, as higher shape variability and complexity is expected.

11. Conclusion

In this paper we investigated the temporal behavior of communication flows in IP networks. We defined a time activity feature vector that captures the temporal behavior of flows. The analysis of the feature vectors by clustering algorithms discovered seven time-activity footprints, namely T1, T2, T3, T4, I1, U1 and U2, embracing all together 95.3% of all flows. Time-activity patterns detected massive events that showed well-delimited shapes in the time-activity expression and were bound to the following communication phenomena:

- T1 identified TCP unsuccessful connection attempts and, eventually, abnormal TCP scanning.
- T2 identified a TCP Tsunami SYN Flood attack and answers to UDP DNS queries within a specific packet size range.
- T3 identified TCP horizontal scans and UDP horizontal scans with no payload. In a lower proportion, TCP flows from scanned hosts.
- T4 identified servers rejecting TCP reconnection attempts, retransmitted SYN from waiting clients and overloaded servers suffering DoS attacks.
- I1 identified ICMP ANT probing requests and replies. Also some UDP scans.
- U1 identified mostly UDP DNS resolutions and, at a minor rate, bittorrent activity.
- U2 identified UDP horizontal scanners and, at a lower rate, UDP DNS resolutions.

Footprints showed that flows were mostly isolated, one-time bursts of few packets (usually only one). Thus, in terms of flows, actual data transmissions or conversations in prolonged flows were the exception. The fact that the one-time burst flows fall into several well defined clusters also shows that clustering according to time-activity patterns provides a valuable method to identify and classify different types of malicious traffic.

The combination of time-activity flow representation with cluster analysis has shown itself to be a suitable tool for filtering traffic and detecting dominant events in IP networks, requiring a minimal inspection of packets. The proposed methodology even detected a misconfiguration problem in one of the MAWI traffic monitors. Future work will focus on: a) the time-activity characterization (footprints) of common, specific network operations; and b) the use of computer clusters (i.e., high performance distributed computing) to analyze traffic from a 1-to-3 year dataset based on the discovered seven patterns.

[10] We refer the interested reader to [41] to probe into performance measurements for classification.

Acknowledgements

We would like to show our gratitude to Prof. Kenjiro Cho and the WIDE Project for their fast responses and support and for making Internet traffic datasets available, which is a very valuable contribution to network research nowadays.

References

[1] 2014 Global Report on the Cost of Cyber Crime, Technical Report, Ponemon Institute, 2014.
[2] Y. Lee, Y. Lee, Toward scalable internet traffic measurement and analysis with hadoop, SIGCOMM Comput. Commun. Rev. 43 (1) (2012) 5–13.
[3] Global Internet Phenomena Spotlight: Encrypted Internet Traffic, Technical Report, Sandvine, 2015.
[4] J.M. Butler, Need for Speed: Streamlining Response and Reaction to Attacks, Technical Report, SANS Institute, 2015.
[5] A. Dainotti, A. Pescape, K. Claffy, Issues and future directions in traffic classification, Netw. IEEE 26 (1) (2012) 35–40.
[6] K. Anagnostakis, S. Ioannidis, S. Miltchev, M. Greenwald, J. Smith, J. Ioannidis, Efficient packet monitoring for network management, in: Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP, 2002, pp. 423–436.
[7] M. Fomenkov, K. Keys, D. Moore, K. Claffy, Longitudinal Study of Internet Traffic in 1998-2003, in: Proceedings of the Winter International Synposium on Information and Communication Technologies, in: WISICT '04, Trinity College Dublin, 2004, pp. 1–6.
[8] J. Charzinski, HTTP/TCP connection and flow characteristics, Performan. Eval. 42 (2–3) (2000) 149–162.
[9] N. Brownlee, One-way traffic monitoring with iatmon, in: Proceedings of the 13th International Conference on Passive and Active Measurement, in: PAM'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 179–188.
[10] H. Wu, M. Zhou, J. Gong, Investigation on the IP flow Inter-Arrival Time in large-scale network, in: Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on, 2007, pp. 1925–1928.
[11] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, C. Diot, A pragmatic definition of elephants in internet backbone traffic, in: Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment, in: IMW '02, ACM, New York, NY, USA, 2002, pp. 175–176.
[12] S. McCreary, K. Claffy, Trends in wide area IP traffic patterns, Technical Report, CAIDA, 2000.
[13] K. Cho, K. Mitsuya, A. Kato, Traffic data repository at the wide project, in: Proceedings of the Annual Conference on USENIX Annual Technical Conference, in: ATEC '00, USENIX Association, Berkeley, CA, USA, 2000.
[14] A. Kato, J. Murai, S. Katsuno, An internet traffic data repository: The architecture and the design policy, in: INET'99, 2012.
[15] P. Barford, D. Plonka, Characteristics of network traffic flow anomalies, in: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, in: IMW '01, ACM, New York, NY, USA, 2001, pp. 69–73.

Please cite this article as: F. Iglesias, T. Zseby, Time-activity footprints in IP traffic, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.012

[16] M. Lee, T. Shon, K. Cho, M. Chung, J. Seo, J. Moon, An approach for classifying internet worms based on temporal behaviors and packet flows, in: D.-S. Huang, L. Heutte, M. Loog (Eds.), Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, volume 4681 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 646–655.
[17] V. Bandara, A. Pezeshki, P. Anura, Modeling spatial and temporal behavior of Internet traffic anomalies, in: IEEE 35th Conference on Local Computer Networks (LCN), 2010, pp. 384–391.
[18] F.I. Vazquez, T. Zseby, Modelling IP darkspace traffic by means of clustering techniques, in: IEEE Conference on Communications and Network Security (CNS), 2014.San Francisco, USA
[19] K. Xu, Z.-L. Zhang, S. Bhattacharyya, Profiling internet backbone traffic: Behavior models and applications, SIGCOMM Comput. Commun. Rev. 35 (4) (2005) 169–180.
[20] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, SIGCOMM Comput. Commun. Rev. 35 (4) (2005) 217–228.
[21] CAIDA, A day in the life of the internet (DITL), Last Modified: Wed Jul-6-2011.
[22] RFC 7011 - Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, Technical Report, Internet Engineering Task Force (IETF), 2013.
[23] RFC 5103 - Bidirectional Flow Export Using IP Flow Information Export (IPFIX), Technical Report, Internet Engineering Task Force (IETF), 2008.
[24] T.A. Lab, ANT censuses of the internet address space, Last Consulted: Aug-2015.
[25] L. Quan, J. Heidemann, Y. Pradkin, Detecting Internet Outages with Precise Active Probing (extended), Technical Report, USC/Information Sciences Institute, 2012.ISI-TR-2012-678b
[26] H.-J. Mucha, H.-G. Bartel, J. Dolata, Effects of data transformation on cluster analysis of archaeometric data, in: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker (Eds.), Data Analysis, Machine Learning and Applications, Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, 2008, pp. 681–688.
[27] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2-3) (2001) 107–145.
[28] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Prez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recog. 46 (1) (2013) 243–256.
[29] Y. Jung, H. Park, D.-Z. Du, B. Drake, A decision criterion for the optimal number of clusters in hierarchical clustering, J. Global Optimization 25 (1) (2003) 91–111.
[30] R. Krishnapuram, J. Kim, A note on the Gustafson-Kessel and adaptive fuzzy clustering algorithms, Fuzzy Syst. IEEE Trans. 7 (4) (1999) 453–461.
[31] F. Klawonn, Fuzzy clustering: insights and a new approach, Mathw. Soft Comput. 11 (2-3) (2004) 125–142.
[32] R.N. Dave, R. Krishnapuram, Robust clustering methods: a unified view, Trans. Fuzzy Syst. 5 (2) (1997) 270–293.
[33] R. Babuka, P. van der Veen, U. Kaymak, Improved covariance estimation for Gustafson-Kessel clustering, in: Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on, 2, 2002, pp. 1081–1085.
[34] H. Liu, S. Shah, W. Jiang, On-line outlier detection and data cleaning, Comput. Chem. Eng. 28 (9) (2004) 1635–1647.
[35] TShark, (https://www.wireshark.org/docs/man-pages/tshark.html).
[36] Python, (https://www.python.org/).
[37] Perl, (https://www.perl.org/).
[38] MATLAB, (https://www.mathworks.de/products/matlab/).
[39] ERT Threat Alert – Tsunami SYN Flood Attack, Technical Report, Radware, 2014.
[40] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, Pattern Anal. Mach. Intell. IEEE Trans. 27 (8) (2005) 1226–1238.
[41] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437.
[42] T. Cover, P. Hart, Nearest neighbor pattern classification, Inf. Theory, IEEE Trans. 13 (1) (1967) 21–27.

Please cite this article as: F. Iglesias, T. Zseby, Time-activity footprints in IP traffic, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.012

Félix Iglesias Vázquez was born in Madrid, Spain, in 1980. He received the Ph.D. degree in technical sciences in 2012 from TU Wien, where he currently holds a University Assistant position. He has worked on fundamental research and project development for diverse Spanish and Austrian firms, and lectures in the fields of electronics, physics and automation. His research interests include machine learning, data analysis and network security.

Tanja Zseby is a professor of communication networks in the Faculty of Electrical Engineering and Information Technology at TU Wien. She received her Dipl.-Ing. degree in electrical engineering and her Ph.D. (Dr.-Ing.) from Technical University Berlin, Germany. Before joining TU Wien she led the Competence Center for Network Research at the Fraunhofer Institute for Open Communication Systems (FOKUS) in Berlin and worked as visiting scientist at the University of California, San Diego.

Please cite this article as: F. Iglesias, T. Zseby, Time-activity footprints in IP traffic, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.012