A deeper look at IP-ID behavior in the Wild

Dario Rossi, Flavia Salutari

***Abstract*—Originally used to assist network-layer fragmentation and reassembly, the IP identification field (IP-ID) has been used and abused for a range of tasks, from counting hosts behind NAT, to detect router aliases and, lately, to assist detection of censorship in the Internet at large. These inferences have been possible since, in the past, the IP-ID was mostly implemented as a simple packet counter: however, this behavior has been discouraged for security reasons and other policies, the use of random values, have been suggested. In this study, we propose a framework to classify the different IP-ID behaviors using active probing from a single host. Despite being only minimally intrusive, our technique is significantly accurate (99% true positive classification) robust against packet losses (up to 20%) and lightweight (few packets suffices to discriminate all IP-ID behaviors). We then apply our technique to an Internet-wide census, where we actively probe one alive target per each routable /24 subnet: we find that the majority of hosts adopts a constant IP-IDs (39%) or local counter (34%), that the fraction of global counters (18%) significantly diminished, that a non marginal number of hosts have an odd behavior (7%) and that random IP-IDs are still an exception (2%). We believe that these findings, together with the datasets we release, can provide some support for works relying on a specific implementation of the IP-ID and, more generally, they can be instrumental for researchers operating in the field of network measurements, by providing them an updated picture of the Internet-wide adoption of the different known IP-ID implementations.**

***Index Terms*—IP-ID, IPv4, Supervised Classification.**

I. Introduction

The IP identification (IP-ID) is a 16 (32) bits field in the IPv4 (v6) header [25]. Originally, along with the fragment offset, the IP-ID was used to assist packet segmentation and reassembly and it was unique per each combination of source, destination and protocol. Yet, with technology evolution and the adoption of the MTU path discovery [22], IP fragmentation becomes less common nowadays, so that the last normative reference [30] allows IP-ID of atomic datagrams to be be non-unique. As a consequence, IP-ID fields values are determined by the specific implementation of the Operating System [23]. In particular, the majority of research work focus their attention on the *global* counter implementation, which used to be the most common implementation about a decade ago [31]. However, due to recent evolution of the standards [11], [30], a wider range of behaviors can be expected nowadays. Over time, different behaviors have been observed such as global and per-flow counters, pseudo-random sequences and constant values [2], as well as odd behaviors such as those due to load balancing [6] middleboxes, or host implementations using the wrong endianness [23]. Given that some of the above implementations maintain state at the IP level, the IP-ID field has been of invaluable help to infer a wealth of information concering the network. Particularly, by leveraging inference from global IP-ID implementation, researchers have been able to count the numer of hosts behind NATs [2], [23], or even assess the traffic they generate [6], [15] and finally expose censorship in the Internet [4], [20], [23], [24].

Given this context, and in particular the emergence of new IP-ID behaviors, it is important to define methods to classify them, as well as using these methods to quantify the prevalence of IP-ID implementation in the current Internet. To summarize our main contributions:

- we design and implement a lightweight methodology to classify the full range of IP-ID behaviors, based on a handful of ICMP packets;
- we carefully validate our method against two datasets comprising the replies from about 1,855 sample hosts, chosen in different manners, for which we build a ground-truth by manual inspection and against multiple synthetic datasets, tailor-made to test robustness against various forms of shortfalls;
- we apply the methodology to an Internet-wide campaign, where we classify one alive target per each routable /24 subnet, gathering a full blown picture of the IP-ID adoption in the wild.

Specifically, whereas the global counter (18% of occurrencies in our measurement) implementation was the most common a decade ago [31], we find that other behaviors (constant 34% and local counter 39%) are now prevalent. We also find that security recommendations expressed in 2011 [11] are rarely followed (random, 2%). Finally, our census quantifies a non marginal number of hosts (7%) showing evidence of a range of behaviors, that can be traced to poor or non-standard implementations (e,.g., bogus endianness; non-standard increments) or network-level techniques (e.g., load balancing, or exogenous traffic intermingled to our probes confusing the classifier). To make our findings useful to a larger extent, we make all our dataset and results available at [26].

The paper is structured as follows: Sec. II discusses the related work. Sec. III describes the methodology and illustrates the workflow and the datasets involved. In Sec. IV we show the performance of the *supervised classification* approach chosen in the following order: system validation, robustness assessment and probing overhead analysis. Sec. V presents the results of the classifier when operated in the wild and put in perspective the findings obtained with those of the previous works. Finally, Sec. VI summarizes the main outcomes and concludes the paper.

II. Background and related work

A. Normative reference

The IP identification (IP-ID) field identifies the unique fragments of a packet and it is used to handle the re-assembling

TABLE I
SUMMARY OF RELATED WORK

Work	Year	Features	Census	Classes Breakdown (%)	Methodology	Scope of the work
[21]	2003	ΔIP-ID	no (only 5000 target routers)	70% global, remaining 30% between constant (equal to 0) and counters with increment by 2.	Analysis of replies to active probing (ICMP requests)	Packet reordering and losses diagnosis.
[6]	2005	Δ IP-ID	no (50 target web-servers)	38% global	Analysis of replies to active probing (HTTP requests)	Discover the amount of load balanced servers, measure the traffic generated by a server.
[14]	2013	-	no	57% global, 14% local, 9% constant, 20% *mixed* IP-ID, 1% random/other ([1])	-	Off-path DNS cache poisoning attacks and defense against them through DNSSEC validation.
[24]	2017	IP-ID acceleration	no	16% global	TCP SYN-ACK from multiple vantage points	Reveal Internet censorship.

process. First documented in the early 80s by RFC791 [25] its use has been updated in several RFCs [5], [9], [11], [12], [30], [31]. Whereas [25] does not fully specify the IP-ID behavior (i.e., it only states that each packet must have a unique IP-ID for the triplet of source, destination and protocol), different behaviors (namely Global, Local and Random, illustrated in Fig.1) are detailed in 2006 by RFC4413 [31]. In 2008, RFC5225 [9] observed that some hosts set the IP-ID to *zero*: at the time of [9], this was a not legal implementation as the field was supposed to be unique. Yet, in 2012 [23] observed that the actual IP-ID implementation depends on the specific Operating System (OS) and versions[2]. In 2013, RFC6864 [30] updated the specifications by affirming that the IPv4 ID uniqueness applies to only non-atomic datagrams: in other words, if the don't fragment (DF) bit is set, fragmentation and reassembly are not necessary and hence devices may set the IP-ID to zero. At the same time, concern has been raised about security problems following the predictability of IP-ID sequences [10], [12], [14], [18]. In particular, in 2012 RFC6274 [11] discouraged the use of a global counter implementation for many security issues, such as stealth port scan to a third (victim) host, and in 2016 RFC7739 [12] addressed concerns concerning IPv6-specific implementations. In light of the recent evolution of the standards, a re-assessment of IP-ID usage in the wild is thus highly relevant.

B. IP-ID Classification Breakdown

In the last decade, to the best of our knowledge, few research works have provided a complete picture of the breakdown of the existing IP-IDs behaviors. That is what makes the comparison of the results of this work with the previous ones with the purpose of analysing the temporal changes on the IP-ID popularity an hard task.

Specifically, the sole quantitative assessment of IP-ID behavior over multiple classes dates back to 2013. This is limited to 271 Top Level Domains TLDs probed by [14] (whose main aim is to propose practical poisoning and name-server blocking attacks on standard DNS resolvers, by off-path, spoofing adversaries). In particular, the 2013 study finds 57% global, 14% local, 9% constant, 1% random/other . Additionally, [14] suggests that 20% of DNS TLD exhibit evidence of "two or more sequential sequences mixed up, probably due to multiple machines behind load balancer".

The remaining works concentrate instead on assessing the popularity of just the global implementation being it only the focus of their studies, proving once again the relevance of a Internet-wide list comprising IP addresses generating IP-ID with the aforementioned behavior. Namely, in 2003, [21] reported that 70% (over 5000 probed targets) were using an IP-ID counter (global or local implementation); in 2005, [6] reported that 38% (over 150 hosts) used a global IP-ID; in 2006, [31] affirms the global implementation to be the most common assignment policy (among 3 behaviors).

C. IP-ID Based-Inference

Additionally, the IP-ID has been exploited for numerous purposes in the literature. Notably, IP-ID side-channel information helped to discover load balancing server [6], count hosts behind NAT [2], [23], measure the traffic [6], [15] and detect router alias [3], [17], [29]. More recently, [20] leverages IP-ID to detect router aliases, or infer router up time [4] and to reveal Internet censorship [24], refueling interest in the study of IP-ID behavior. Whereas the above work [2], [6], [15], [24], [29] mostly focus only on the global IP-ID behavior, in this work we not only consider all *expected* IP-ID behavior, but additionally quantify *non-standard* behaviors: in particular, we provide a methodology to accurately classify IP-ID behaviors, that we apply to the Internet at large, gathering a picture of the relative popularity of each IP-ID behavior. In terms of methodologies, authors in [21] use ICMP timestamp and IP-ID to diagnose paths from the source to arbitrary destinations and find reordering, loss, and queuing delay. In [16], the authors identify out-of-sequence packets in

[2]In particular [23] reports Windows and FreeBSD to use a global counter, Linux and MacOS to use local counters and OpenBSD to use pseudo-random IP-IDs.

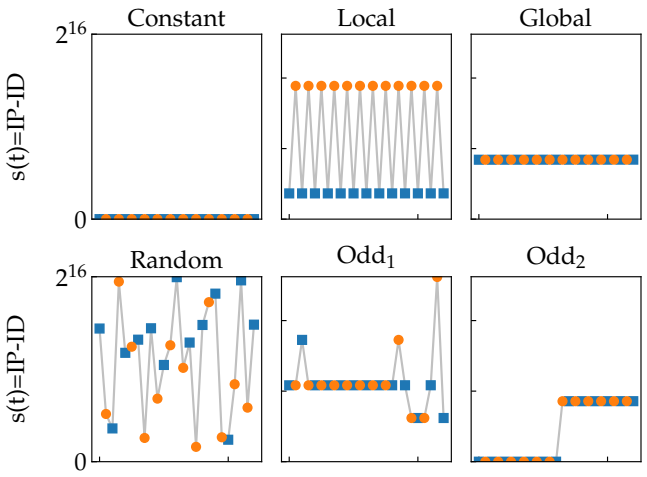

Fig. 1. Illustration of Constant, Local, Global, Random and Odd sequences

TCP connections that can be the result of different network events such as packet loss, reordering or duplication. In [6], they use HTTP requests from two different machines toward 150 target websites, to discover the number of load-balancing server. Authors in [24] use TCP SYN-ACK from multiple vantage points to identify connectivity disruptions by means of IP-ID fields, which then they use as a building block of a censorship detection framework.

Building on our own previous work [27], we leverage ICMP traffic (spoofing IP addresses to craft sequences of packets that are precisely interleaved when they hit the target under observation) to build an accurate, robust and lightweight IP-ID classification technique. In particular, this work extends [27] by providing more details on the experiments run and on the methodology, additional results, such as the sensitivity analysis and the spatial analysis.

III. Methodology

To provide an accurate and comprehensive account of IP-ID behavior in the wild, we need (i) a reliable classifier, able to discriminate among the different typical and anomalous IP-ID behaviors. At the same time, to enable Internet coverage, (ii) the classifier should rely on features with high discriminative power, extracted from the data gathered through an active probing technique that is as lightweight as possible. In this section we illustrate the practical building blocks and their theoretical foundations, that our classification framework builds upon.

A. IP-ID classes

From the host perspective, several IP-ID behaviors are possible as depicted in Fig.1. The image shows the sequences of 25 IP-ID samples sent from 2 different host (orange and blue) where the packets are sent alternatively to the target. The different behaviors depicted are, from left to right: (i) **constant** counters are never incremented (and for the most part are equal to `0x0000`); (ii) **local** or per-host counters that are incremented at each new packet arrival for that flow (mostly by 1 unit, 99.7% of the times in our large scale measurements): as a consequence, while the orange or blue per-host sub-sequences are monotonically increasing, the aggregate sequence alternates between the two; (iii) **global** counters are incremented by 1 unit at each new packet arrival for any flow: thus, the sequence s is monotonically increasing (90.3% of the times by 1 unit, 4.7% by 2 units and 4.6% by 3 units), and the orange or blue per-host sub-sequences are monotonically increasing but at a faster rate (by 2 units); (iv) **random** IP-IDs are extracted according to a pseudo-random number generator. Finally, a special mention is worth for the class of (v, vi) **odd** IP-ID behaviors, that are not systematically documented in the literature and that arise for several reasons (including bugs, misconfiguration, non-standard increments, unforeseen interaction with other network apparatuses, etc.) and for which we report two different samples occurring in real experiments.

B. Active probing

To gather the above described sequences, our measurement technique relies on active probing. We craft a tool able to send and receive ICMP packets, running at two vantage points (VP) with public IP addresses in our campus network. Specifically, we send a stream of N ICMP echo requests packets in a *back-to-back* fashion, which forces the target machine to generate consecutive ICMP echo replies: thus, assuming for the time being that no packet were lost, we gather a stream of N IP-IDs samples for that target. Sending packets back-to-back is necessary to reduce the noise in the IP-IDs stream sequence: if probe packets were spaced over time, the sequence could be altered by exogenous traffic hitting the target (e.g., in case of global counter). As a result, the sequence would depend on the (unknown) packet arrival rate in between two consecutive probe packets, likely confusing the classifier [28]. In this way, the use of back-to-back packets reduces as much as possible the interference with some possible extra exogenous traffic hitting the same destination, that could otherwise alter the sequences [28]. A second observation is that, whereas a single vantage point may be sufficient to distinguish among constant, random and global counters, it would fail to discriminate between global vs local counters. However, sending packets from two different VPs is not advisable, due to the difficulty in precisely synchronizing the sending patterns so that packets from different hosts alternate in the sequence [21].

Therefore, a better alternative is to receive packets on two different VPs, x and y, but shift the packet generation process to only one of them, as x, and use it as sender: by letting x spoof the address IP_y of the colluding receiver y, it is possible to generate a sequence of *back-to-back packets* that are also *perfectly interleaved* as depicted in Fig.1. Fig.2 shows the scenario in which the experiments are carried out. It provides information about how the hosts are involved and the kind of data collected: there are two receivers but only one real sender, and the information gathered at the two vantage points regards the sequences of IP-IDs generated by the target machine. To validate our assumptions, we carry on additional experiments

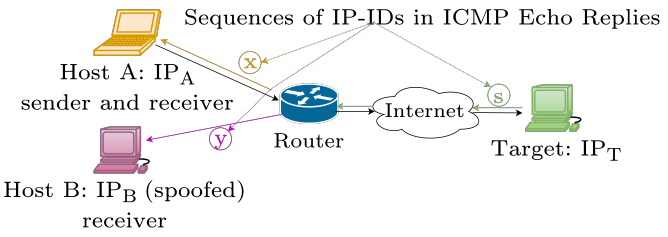

Fig. 2. Scenario in which the active probing is performed: only one sender is used to ease the synchronization of packets generation, whilst both the machines are used to receive and collect the stream of IP-IDs generated at the target machine.

on a preliminary testbed to test the sensitivity of the algorithm to external traffic hitting the target. In these experiments:

- we send UDP CBR traffic with Iperf at $TX_{rate} = 10Mbps$ and vary the packet size over time (in particular decrease), so that we increase the packet rate during the experiment (to control the IP-ID generation);
- in one experiment, we send ICMP Echo Request packets with an inter-packet gap of $\Delta t_{interpacketgap} = 10ms$ and collect the IP-ID sequence x, for which we derive the derivative series (gray color line);
- in the other experiment, we send ICMP packets back-to-back and again measure the growth of IP-ID in the sequence (red color line).

Even though the experiments are simple, the results are very telling: the plots in Fig. 3 show the derivative of the sequence of IP-IDs x', which basically just counts the amount of exogenous packets in between two consecutive ICMP probes, in both scenarios of the experiments. For instance, when packets are 100 Bytes long, in $\Delta t_{interpacketgap} = 10ms$ it is expected to have $\frac{TX_{rate} \cdot \Delta t_{interpacketgap}}{packetsize} = 125$ packets slipping in between two probes, which actually happens. This would clearly jeopardize the classifier. Conversely, in the experiments carried out in our lab, back-to-back packets leave no possibility to the other UDP packets to intermingle and confuse the classifier. These experiments suggest that sending packets back-to-back is a good strategy, although we do not feel results to be conclusive for all the devices available in the network (e.g. router, setup, shaper, etc.). However, even in case the reality was not as nice as our lab results (which is likely to be the case), at the same time this affects at most some of the *odd* behaviors, which already are a tiny (7%) fraction of the overall cases. Indeed, it is very unlikely that the amount of real traffic is so perfectly varying between probes to erroneously confuse a classifier to believe that a *global* sequence is a *random* one just due to exogenous traffic. Very high information entropy of those sequences is not a side-product of some variable traffic, but truly coming from a random number generator (it is pretty well known that is hard to generate good pseudo-random sequences, and the arrival rate is surely not a source of perfect entropy).

To overcome reordering, packet loss and duplication events, we additionally control the sequence number in the stream of generated probe packets.

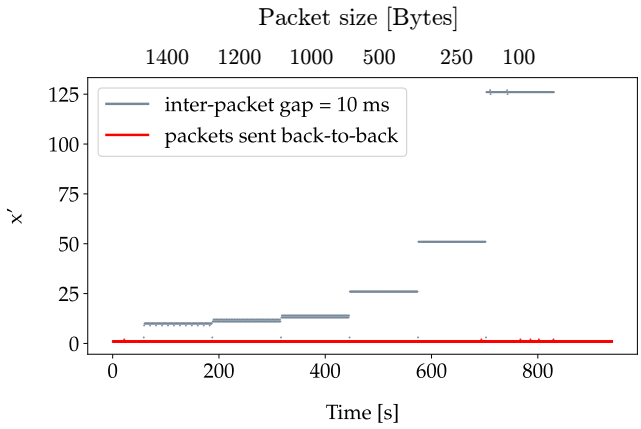

Fig. 3. Sensitivity analysis to external traffic: derivative of the sequence of IP-IDs x' in the two different scenarios

TABLE II
TABULATED EXPECTED VALUES FOR SELECTED FEATURES

Feature	Constant	Local	Global	Random	Odd
$H(x)$	0	$\log_2 \frac{N}{2}$	$\log_2 \frac{N}{2}$	$\leq \log_2 \frac{N}{2}$	-
$H(s)$	0	$\leq \log_2 N$	$\log_2 N$	$\leq \log_2 N$	-
$H(x')$	0	0	0	$\leq \log_2 \frac{N}{2}$	-
$H(s')$	0	1	0	$\leq \log_2 N$	-
$\mathbb{E}[x']$	0	1	2	$\frac{(2^{16}-1)}{2}$	-
σ_x	0	$\sqrt{\frac{(N^2-4)}{48}}$	$\sqrt{\frac{(N^2-4)}{12}}$	$\frac{(2^{16}-1)}{\sqrt{12}}$	-
σ_s	0	$\leq \frac{(2^{16}-1)}{\sqrt{12}}$	$\sqrt{\frac{(N^2-1)}{12}}$	$\frac{(2^{16}-1)}{2}$	-
σ'_x	0	0	0	$\frac{(2^{16}-1)}{\sqrt{12}}$	-
σ'_s	0	$\lvert x_1 - y_1 - \frac{1}{2} \rvert$	0	$\frac{(2^{16}-1)}{\sqrt{12}}$	-

C. Features Definition

As anticipated, to build a robust classifier we need to *manually* define a set of tailor-made features able to discriminate among the different IP-IDs implementations. The experiment and the measurements can be formalised as follows: we send N packets to a given target t, with the source address field alternating between consecutive requests, whose replies are sent back to our two vantage points x and y: we indicate with s the aggregated sequence comprising the N IP-IDs sent back by t, as we receive it at the edge of our network[3]. By abuse of language, we indicate with x and y the sub-sequences (each of length $N/2$) of IP-IDs, sent back by t and received by the homonyms host. From these sequences x, y and s we further construct derivative series x', y' and s' by computing the discrete differences between consecutive IP-IDs (i.e., $x'_i = x_i - x_{i-1}$). We summarize these series with few scalar features by computing the first

$$\mathbb{E}[X] = \frac{1}{N} \cdot \sum_{i}^{N} x_i \tag{1}$$

[2]Due to the rounding done by the authors [14], the sum of all the percentages is 101%

[3]Notice that packet losses and reordering may let us receive less than N packets, or receive packets in a slight different order than what sent by the target.

and second moments

$$\sigma = \sqrt{\mathbb{E}[X^2] - (\mathbb{E}[X])^2} \tag{2}$$

of the IP-ID series, as well as their information entropy,

$$H(X) = \mathbb{E}\big[\mathbb{I}(X)\big] = -\sum_{i}^{N} p_i \log_2 p_i \tag{3}$$

which is defined as the expected value of the information content $\mathbb{I}(X) = \log_2 \frac{1}{p_i}$, where p_i is the provability that the discrete random variable X takes the x_i value.

Specifically, we consider the mean $\mathbb{E}[X]$ of the derivative series $x\prime$ and $y\prime$, the entropy $\mathbb{H}(X)$ and the standard deviation σ of s,x and y and of their derivatives s', x' and y' . Actually, for each feature we can derive an *expected value* in the ideal[4] case (so that no expected values is reported for the odd class) that we summarize in Tab.II. For the sake of brevity, we report in Tab.II only once the expectations of the features of the subsequences x and y, given that they are conceptually equivalent. Intuitively, we expect the mean of the constant sequence to be unknown, but that of its derivative to be null. Similarly, the derivative of a global counter would have a value of 1 (2) for the aggregate sequence s (subsequences x and y). The entropy of the sequence is expected to increase from the minimum of a constant sequence equal to

$$\mathbb{H}(X) = -1 \log_2(1) = 0 \tag{4}$$

to the maximum of

$$\mathbb{H}(X) = -N \cdot \frac{1}{N} \log_2 \frac{1}{N} = \log_2 N \tag{5}$$

occurring when all the N elements of the series are different. Consequently, by considering the global and local implementations, we can observe that the entropy for the sequences x and y of length $\frac{N}{2}$ is expected to be maximum $\mathbb{H}(x_{global}) = \mathbb{H}(y_{global}) = \log_2 \frac{N}{2}$. Consequently, in the global implementation, the sequence s is made up of two not-overlapping sequences, leading to an expected maximum entropy of $\mathbb{H}(s_{global}) = \log_2 N$. Differently, in the local implementation this is true only when the two counters do not overlap, otherwise this value remains only an upper bound. A similar observation can be done for the entropy expectations for the random sequences, in which the presence of duplicate values would reduce the entropy. For the local implementation, the sequences x', y', derivatives of two independent counters, are constant thus the entropy, as said, is expected to be 0. On the other hand, the derivative s' of the aggregate sequence s is made up of two alternating values, corresponding to the two offsets:

$$s'_{local}(n) = \begin{cases} \theta_1 = y_1 - x_0 & \text{if } n \text{ even} \\ \theta_2 = x_2 - y_1 & \text{if } n \text{ odd} \end{cases} \tag{6}$$

[4]Sequences from well behaving hosts that have no software bug or malicious behavior, and that are neither affected by losses nor reordering

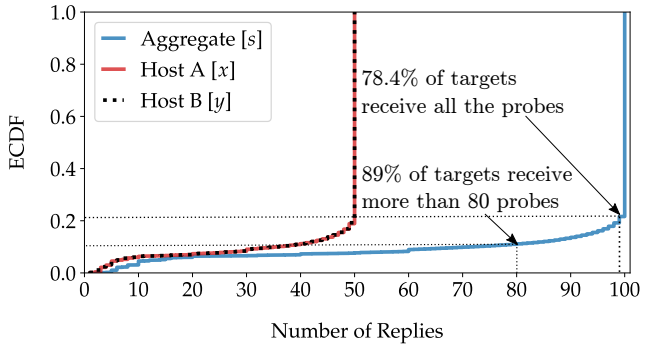

Fig. 4. Internet campaign: ECDF of the number of packet replies

Both θ_1 and θ_2 are repeated for $\frac{N}{2}$ times, so each one occurs with a probability of $\frac{1}{2}$. The entropy becomes:

$$\mathbb{H}(s'_{local}) = -2 \cdot \frac{1}{2} \log_2 \frac{1}{2} = \log_2 2 = 1 \tag{7}$$

Conversely, being the expected derivative sequence of a global counter always equal to $s'_{global} = 1$, as a result the entropy becomes $\mathbb{H}(s'_{global}) = 0$.

In a similar way, the other expectation values can be easily derived by analogy.

D. Datasets

In this work, we collect four different datasets, that we use in the different stages of the work alternatively to make the classifier learn the classification function, i.e. as training dataset, and to evaluate performances as testing dataset.

1) Large scale census $\mathcal{L}$: The first dataset is made up of real measurements coming from the large scale measurement campaign and includes the replies coming from a subset of a hitlist of alive IP addresses. We avoid putting stress on the infrastructure carrying a full Internet census: as we aim at providing an accurate picture of the *relative* popularity of IP-ID implementations on the Internet, it suffices to collect measurements for a large number of targets, namely 1 alive IP/32 host per each /24 prefix. For this reason, for the targets selection, we rely on the public available hitlist regularly published by [13], comprising 16 millions of targets IP/32. The hitlist contains targets for all /24, including those who have never been replying to the probing: excluding them from our target list, leaves us with approximately 6 millions of potential targets. To reduce the amount of probe traffic, we decide to be conservative: we preliminary probe the 6 millions potential targets sending two ICMP echo requests, and include in our final target list the approximately 3,2 million responsive hosts (in line with [8], [32]). We send a batch of N=100 back-to-back probe packets to each target, but otherwise probe at a low average rate, so that we complete a /24 census in about 3 days. Fig.4 shows the empirical cumulative distribution function (ECDF) of received packets at our VPs. We observe that we receive almost all the replies from most of the targets: the 90% (80%) of the targets answer to more than 40 (all) packets per each host, corresponding to a 20% (0%) loss scenario. A large

TABLE III
SUMMARY OF THE DATASETS.

Name	Type	Description	Properties	Size [Targets]	URL
$\mathcal{L}$	Real Measurements	Large scale measurements dataset comprising the IP-ID sequences received from the portion of hitlist [13] providing response rate $\geq 80\%$	Presence of presence of odd behaviors of the IP-ID, possibility of losses or out-of-order packets	2,5 M	[26]
$\mathcal{G}$	Real Measurements	Manually labeled dataset containing the IP-IDs contained in the replies of a set of IP addresses sampled uniformly from the hitlist to guarantee class balance	Targets chosen to provide IP-prefix level and class balance, presence of odd behaviors, used for training and classification of $\mathcal{L}$	2 k	[26]
$\mathcal{G}'$	Real Measurements	Manually labeled dataset containing the IP-IDs from the replies of a set of IP addresses where 75% of it belong to the same IP/8 subnet	Targets chosen to provide IP-prefix level imbalance, presence of odd behaviors, used for validation of performances	2 k	[26]
$\mathcal{S}_{ideal}$	Synthetic	Dataset manually designed to intentionally contain the four possible IP-ID implementations in the ideal case evenly distributed emulating the replies collected through real measurements	Lossless, absence of odd behaviors, used for validation of performances	20 k	[26]
$\mathcal{S}_{lossy}$	Lossy Synthetic	Dataset manually designed to intentionally contain the four possible IP-ID implementations spoiled with four different flavour of losses ($\mathcal{S}_{lossy} = \cup(\mathcal{S}_{unif.}, \mathcal{S}_{hole}, \mathcal{S}_{extr.}, \mathcal{S}_{equi.})$) evenly distributed	Lossy, absence of odd behaviors, used for testing resilience to losses	20 k	[26]
$\mathcal{S}_{reorder}$	Synthetic	$\mathcal{G}$ dataset spoiled when of 20% of each IP-ID sequence is intentionally randomly swapped	Used for testing resilience to sequence alteration due to out-of-order packets	20 k	[26]

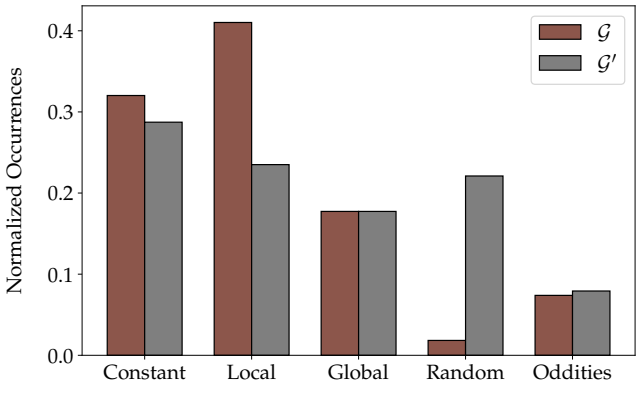

Fig. 5. Manual Ground Truth: Normalized classes occurrences for the training datasets $\mathcal{G}$ and $\mathcal{G}'$

plateau in the CDF also indicates that the distribution is bimodal, i.e., the remaining hosts generally reply with very few packets (e.g., 10 or less per each VP or over 90% loss rate). This suggests that future campaigns could be safely conducted with a smaller $N' < N$. To provide accurate classification results, in light of our robustness analysis done with synthetic dataset and whose results are shown in Sec.IV-B, we limit our attention to the 2,588,148 hosts for which we have received at least $N = 80$ packets.

2) Ground Truth $\mathcal{G}$ and $\mathcal{G}'$: The second real dataset is $\mathcal{G}$, made of IP-ID sequences for which we manually construct a ground truth. For this purpose, we extract the replies from a subset of targets of $\mathcal{L}$ which satisfy some pre-established requirements. We include in this dataset only the 1,855 hosts from which we receive 100% of the replies, and perform the manual inspection of each of the sequences. We repeat the process twice, with two very different choices of the ground-truth datasets: $\mathcal{G}$ sampled uniformly from the hitlist paying attention to guarantee class balance and $\mathcal{G}'$ where about 75% samples belong to the same IP/8 subnet. Interestingly, when performing the manual labelling, we find a small but non marginal fraction (about 7%) of sequences that are hard to classify: a deeper investigation reveals these odd behaviors to be due to a variety of reasons – including per-packet IP-level load balancing, wrong endianness, non standard increments in the global counter, etc. While we cannot completely rule out interference of exogenous traffic altering our IP-ID sequences, lab experiments suggest that the use of back-to-back packets lessen its impact, as described before in Sec.III-B. Nevertheless, these samples provide a useful description of the odd class, that would otherwise have been difficult to define. In Fig. 5 we report the breakdowns of the two datasets $\mathcal{G}$ and $\mathcal{G}'$.

3) Syntethic Datasets: In order to assess the robustness of our classifier against packet losses, we rely on two more datasets which are made up by synthetic sequences, from which we can derive the features useful in the classification process. While for simple loss patterns (e.g., uniform i.i.d. losses) it is still possible to analytically derive expected values in closed form, for loss models where losses are correlated, this becomes significantly more difficult. As such, we opt for an experimental assessment of classification accuracy in presence of different synthetic loss models, that we apply to synthetic ideal sequences contained in dataset $\mathcal{S}_{ideal}$ by purposely discarding a part of the sequences. Specifically, we consider: (i) a **uniform** i.i.d. loss model; (ii) a **hole** model where, starting from a random point in the sequence, 20% of consecutive samples are removed; (iii) an **extreme** model where we remove 20% of the initial values (or equivalently the final 20% of the sequence); and finally (iv) an **equidistant**

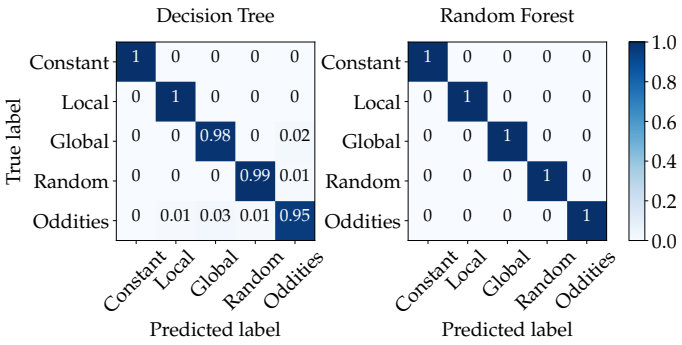

Fig. 6. Validation: Confusion Matrix of 20-fold validation over $\mathcal{G}$ done both with Decision Tree and Random Forest Classifiers

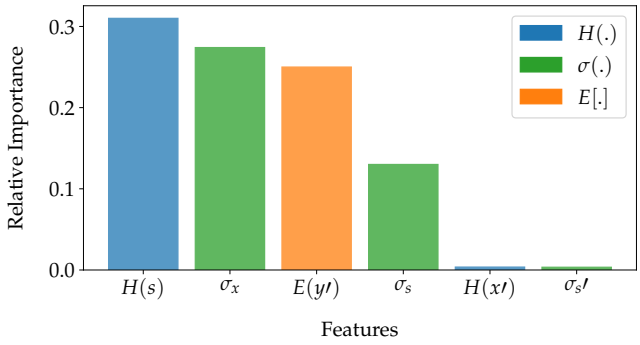

Fig. 7. Validation: Relative importance for the most useful features of the classifier.

model where losses start at a random point and are equally spaced over the sequence. We apply these loss models to obtain a synthetic lossy dataset $\mathcal{S}_{lossy}$. Specifically, for each loss model we generate 5,000 loss sequence pattern, for an overall of 20,000 test cases. In order to deeper investigate the reordering phenomena effect on the performances of the classifier, we manually disrupt the sequences contained in $\mathcal{G}$. Specifically, we impose the swapping of 20% on the IP-IDs contained in the series x, y collected for each IP address in $\mathcal{G}$ and build a new rigged dataset $\mathcal{S}_{reorder}$.
A summary of all the datasets with their description and properties is shown in Tab.III.

IV. IP-ID Classification

From the values tabulated in Tab.II, we expect classifiers that use this set of features to be able to fully discriminate the set of IP-ID well-defined behaviors under ideal conditions. However, as we shall see, unexpected behavior may arise in the Internet, due to a variety of reasons, which are hard to capture in general. We thus opt for a *supervised classification* approach, which allows to learn a predictive model with decision trees (DTs), based on the above features. Specifically, we resort to the Classification And Regression Trees (CART) [19], that builds trees having the largest information gain at each node. DTs are part of the *supervised* machine learning algorithms, and infer a classification function from a *labeled* training dataset, that we have manually built and that is useful for training and validation purposes. Additionally, we investigate to what extent the classifier is robust against losses and reordering, and finally assess the minimum number of samples N needed to achieve a reliable classification.

A. Classification accuracy and validation

We first train and validate our classifier using the the real dataset $\mathcal{G}$ of IP-ID sequences for which we have manually constructed a ground truth. Note that, for the moment we train the classifier only over the dataset $\mathcal{G}$, but later we will show the independence of the model from this choice.

We assess the classification accuracy over $\mathcal{G}$ with a 20-fold cross-validation, whose results are reported in Fig. 6 as a confusion matrix: we can observe that the classifier is extremely accurate, with 100% true positive in the constant and local classes, 99% for the random and 98% for the global class. The worst case is represented by 95% true positive for the odd class (that represent only 7% of the samples): these very few misclassifications are erroneously attributed to local, global or random classes, and additional series definition (e.g., to compensate for wrong endianness) could help reducing if needed. For completeness, in order to compare the classification methodology used with another one, we compare the results obtained with the CART Decision Tree algorithm with the ones done with the Random Forest Classification. Results, shown in Fig. 6, again as a confusion matrix, show that the small misclassification gaps introduced by the Decision Tree are fully filled when using a Random Forest Classifier, which leads to 100% classification accuracy for all the classes.

Additionally, Fig. 7 depicts the importance for the most useful features of the classifier. Four main takeaways can be gathered from the picture: first, just four features are necessary for a full discrimination, which is reasonable as the cardinality of the classes to discriminate is small; second, as expected features that measure the dispersion (entropy and standard deviation) are prevalent; third, both original and derivative sequences are useful in the detection; fourth, subsequence metrics are highly redundant (i.e., $H(x) = H(y)$, $\sigma_x = \sigma_y$, etc.).

B. Robustness

It is fundamental to test the robustness of the features to losses in a controlled scenario, in order to emulate the real measurements, in which events such as packet losses or out-of-order arrivals are not so rare. For the previously shown six features we evaluate their values in the lossy synthetic $\mathcal{S}_{lossy}$ sequences and tabulate the results averaged over the dataset, respecting the IP-ID and loss type partitioning, in order to compare with the ones of evaluated for the lossless sequences in $\mathcal{S}_{ideal}$. In Tab. IV we report those values evaluated for the simulated local implementations. The columns represent the different cases in which the features are evaluated, lossless dataset $\mathcal{S}_{ideal}$, and lossy $\mathcal{S}_{lossy}$ with uniform random, hole, extremal and equidistant losses. As a whole, results obtained with the synthetic dataset $\mathcal{S}_{lossy}$ do not significantly diverge from the ones deriving from $\mathcal{S}_{ideal}$ proving the strength of

TABLE IV
FEATURES VALUES FOR BOTH LOSSLESS AND LOSSY SYNTHETIC DATASET $\mathcal{S}_{lossy}$ WITH 20 % LOSSES - LOCAL IMPLEMENTATION CASE OF IP-ID.

	$\mathcal{S}_{ideal}$	$\mathcal{S}_{lossy}$			
Feature	Lossless	Uniform	Hole	Extremal	Equidistant
$H(s)$	6.64	6.64	6.64	6.64	6.64
$H(x')$	0	0.84	0.17	0	0.78
$\mathbf{E}[y']$	1	1.25	1.25	1	1.26
$\sigma(x)$	32.64	38.75	29.68	18.02	20.92
$\sigma(s)$	$10.97e^3$	$11.09e^3$	$1072e^3$	$11.03e^3$	$11.05e^3$
$\sigma(s')$	$16.29e^3$	$16.01e^3$	$16.6e^3$	$16.15e^3$	$16.4e^3$

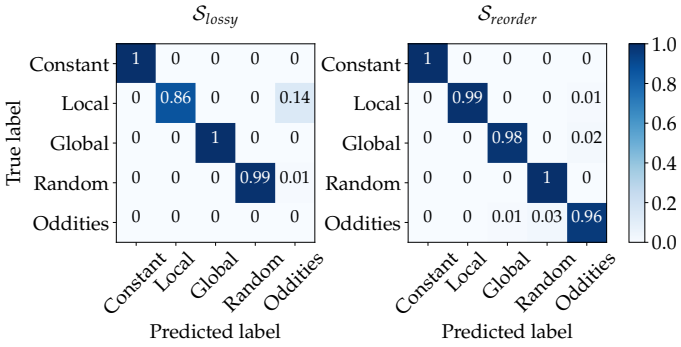

Fig. 8. Robustness: (left) Confusion Matrix of a classifier trained over the real lossless dataset $\mathcal{G}$ and tested over synthetic lossy dataset $\mathcal{S}_{lossy}$ with purposefully injected 20% packet losses on each sequence, (right) Confusion Matrix of a classifier trained over the real lossless dataset $\mathcal{G}$ and tested over the dataset where 20% of each sequence is intentionally randomly swapped $\mathcal{S}_{reorder}$.

the features and their robustness to change and alteration of the original sequences. Specifically, $H(s)$, which turns out to be the most important feature, as shown in Fig. 7, does not vary in presence of any kind of losses, $H(x')$ can vary more depending on the flavour of the loss.

Given this results, we next assess the robustness of the classifier against packet losses, which may introduce distortion in the features. Since, as previously described, the expected values in the ideal conditions are significantly apart, we expect the classifier to be resilient to a high degree of losses. Without loss of generality, we consider an extreme case where only 80 out of 100 samples are correctly received (i.e., a 20% loss rate) by exploiting the lossy synthetic dataset $\mathcal{S}_{lossy}$.

We want to assess the accuracy of the previously validated model, i.e., the one trained on the real lossless dataset $\mathcal{G}$ over $\mathcal{S}_{lossy}$. Results of these experiments are reported in Fig.8 and Fig.9. In particular, the confusion matrix reported in the left side of Fig.8 shows the aggregated results over all loss models: we can observe that most of the classes have a true positive classification of 99% or 100% even in presence of 20% packet losses, and irrespectively of the actual loss pattern.

Additionally, we observe that in the case of the *local class*, only 86% of the sequences are correctly classified, whereas 14% of the local sequences in presence of heavy losses are erroneously classified as being part of the "odd" behavior class. Fig.9 dig further the reasons of this discrepancy, showing that the misclassification mostly happens for the *hole* loss

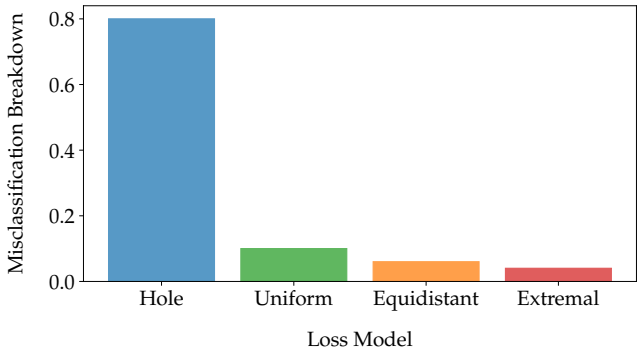

Fig. 9. Robustness: Misclassification breakdown of the (local,odd) (14%) for the different loss models.

model, while in the other cases is a very rare event. Recalling the odd behavior early shown in the plot of Fig. 1, we notice that this model induces a gap in the sequence, which is possibly large enough to be statistically similar to cases such as load balancing, where the sequence alternates among multiple counters. Overall, we find the classifier to be robust to very high loss rates and, with a single exceptions, also invariant to the actual loss pattern – which is a rather desirable property to operate the classifier into a real Internet environment. To investigate the effect of the presence of out-of-order packets received at the vantage point and of the reordering phenomena, we perform again the classification, with the decision tree classifier still trained over $\mathcal{G}$ but tested over $\mathcal{S}_{reorder}$. We use again the confusion matrix as graphical way to highlight the quality of the classification. Results of these experiments are shown in the right matrix of Fig.8: we can observe that reordering does not affect at all constant and random labels classification and that the classifier is strong in recognizing the local and global behaviors leading to respectively 1% and 2% false positive misclassification.

C. Probing Overhead

We finally assess how large the number of samples N needs to be to have accurate classification results. In principle, features tabulated in Fig.II are diverse enough so that we expect high accuracy even for very small values of N.

To assess this experimentally, we take the real lossless dataset $\mathcal{G}$ and only consider that we have at our disposal only $N' < N$ out of the $N = 100$ samples gathered in the experiment. For each value of N', we perform a 20-fold cross validation, training and validating with N' samples. We start from a minimum of $N' = 10$ (i.e., 5 packets per host) up to the maximum of $N = 100$ (i.e., 50 probes per host) samples. Fig.11 clearly shows that accuracy is already very high[5] at 0.95 when $N' = 4$ and exceeds 0.99 when $N = 100$.

[5]Notice that even in the extreme case with as few as $N' = 2$ packets, random and constant classification are correctly labeled, whereas the remaining global vs local cannot be discriminated, yielding to 0.70 accuracy in the $\mathcal{G}$ set.

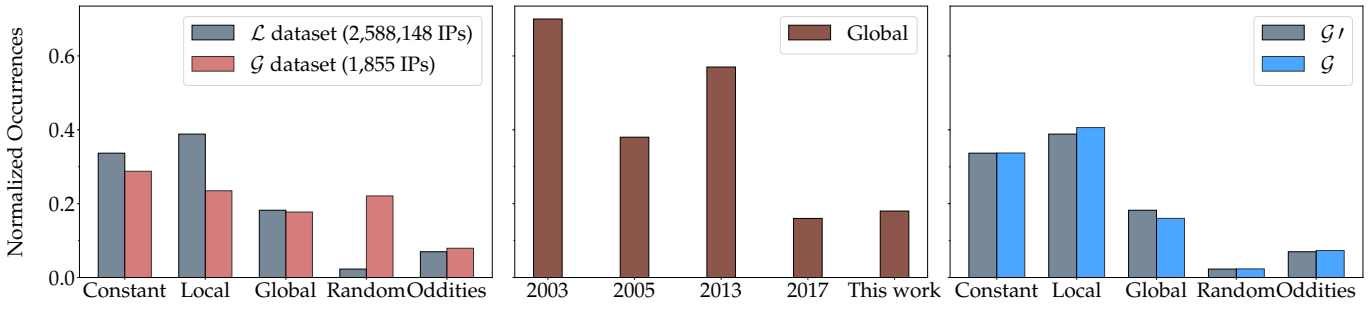

Fig. 10. (a) Internet campaign: Normalized classes occurrences for the training $\mathcal{G}$ and Internet-scale $\mathcal{L}$ dataset; (b) Measured occurrences of Global IP-ID implementations over the years; (c) Breakdown of the classes of $\mathcal{L}$ obtained with both G' and G

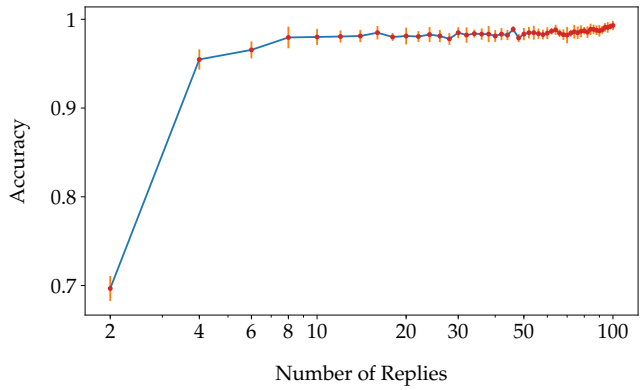

Fig. 11. Probing Overhead analysis: Accuracy as a function of the sample set size

V. Internet Campaign

The last step of the analysis consists in using the previously trained classifier over $\mathcal{G}$ to classify the IP-ID behaviors present in the dataset $\mathcal{L}$. In this section, we firstly detail the way in which we collect the data, then we show the results of the classification and we compare them with ones done in the past. Then, we deeper investigate some aspects to see how different boundary conditions affect classification performances, as the impact of different training set choices or of the number of probe packets on the performances of the classification. Finally, we perform a spatial analysis and we deepen the analysis of odd behaviors.

A. Longitudinal Comparison (over the years)

Finally, we apply our classifier in the wild, specifically on the already mentioned dataset $\mathcal{L}$ in Sec.III-D1, made with the data collected through a large scale Internet measurement campaign. We observe that, while our classifier is able to perform a very accurate classification even with few samples, we need to deal with loss rates, which is unknown a priori. Hence, even though our probing overhead analysis in Sec. IV-C revealed high accuracy for few number of samples, we prefer for the time being to use a simple and conservative approach and select $N = 100$ samples, being very accurate also in presence of very high loss rates. We apply the classification to batches of 100,000 hosts, and for each class c, compute the relative breakdown of the class in that batch $\hat{n}_c = n_c / \sum_i n_i$, evaluating the confidence intervals of $\hat{n}_c$ over the different batches. Results are reported in Fig.10 (a), where we additionally report the breakdown in our $\mathcal{G}$ training set comprising just 1,855 population samples: it can be seen that while $\mathcal{G}$ has no statistical relevance for the census, it is not affected by class imbalance and thus proves to be a good training set.

Results are particularly interesting to put in perspective with current literature knowledge. Specifically, past work [6], [10], [21], [31] consistently reported the global counter to be more widespread: in 2003, [21] 70% ; in 2005, [6] 38%; in 2006, [31] affirms the global implementation to be the most common assignment policy; in 2013, [14] 57%. On the contrary, we find that only 18% (over 2,5 million targets) are still using global counter implementation: this in line with 2017 results that reports slightly more than 16% global IP-IDs [24] (whose main aim is to detect censorship in the Internet). While this decreasing trend, summarized in Fig.10 (b), is possibly affected by the comparably smaller population size of early studies, however we believe this trend to be rooted into OS-level changes in IP-ID policy implementations: e.g., Linux and Solaris, which previously adopted a global counter, for security reasons later moved to a local counter implementation [11].

By comparing our results with the only one providing the occurrences of both the normatives-compliant IP- ID behaviors and some odd practices [14], the 2013 study (our census) finds 57% (18%) global, 14% (39%) local and 9% (34%) constant IP-IDs, which testify of a significant evolution. Additionally, recalling that [14] suggests that 20% of DNS TLD generate *mixed*IP-IDs, we find out that this is much larger than the 7% fraction of the larger "odd" class (including but not limited to load balance) that we find in this work. Finally, despite 2012 recommendations [11], the percentage of random IP-ID sequence was (and remains) limited 1% (2%).

For completeness and in light of what showed in Sec. IV-A, we compare the results obtained with the CART Decision Tree algorithm with the ones done with the Random Forest Classification. From the outcomes reported in Fig.12 we can observe that no statistical difference is present in the two cases.

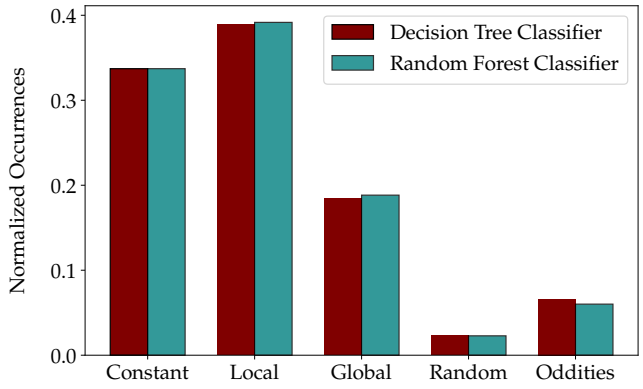

Fig. 12. Breakdown of the classes of $\mathcal{L}$ obtained with both a Decision Tree and a Random Forest Classifier.

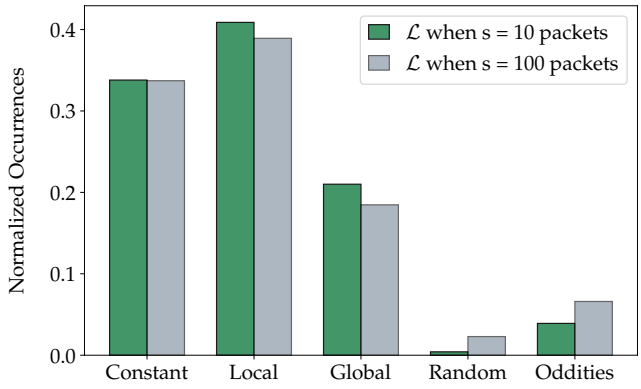

Fig. 13. Normalized classes occurrences for $\mathcal{L}$ and its lighter version when only N=10 packets out of 100 are considered.

B. Sensitivity Analysis

1) Training Set Choice: In order to prove the indipendence of the results from the choice of the training dataset we exploit the second manually validated dataset $\mathcal{G}'$, which satisfies the previously described requirements and it is purposely biased, as it contains 75% of the samples from the same /8, which is something not desiderable from a IP coverage point of view.

We then use these two datasets to classify the IP-ID behaviours in the whole large scale dataset $\mathcal{L}$ covering the all the responsive IP addresses of the full hitlist. Results, shown in Fig. 10 (c) confirms indeed the validity of our methodology since, statistically, there are only slight differences between the occurrences breakdown when the classifier is trained over $\mathcal{G}$ or $\mathcal{G}'$. Both datasets yield to consistent results ensuring the independence of the model from the training dataset and proving that as long as the behaviors are balanced the IP-prefix level imbalance is irrelevant.

2) Lightweight Census: Additionally, we may want to further investigate how the classification results change when we have a fewer number of packets building the IP-ID series that we aim at classify. This is important since we want to avoid injecting useless traffic in the network and, if we find out that we can lead experiments with less samples, we can achieve it. Similarly to what previously described in Sec.IV-C, we take the measurements dataset $\mathcal{G}$ and only consider that we have at our disposal only the first $N' = 10 < N$ out of the N = 100 samples gathered in the full experiment. Given that in this case we are only looking at a small portion of the collected series, we may expect that in this case we can have a loss in terms of amount of *oddities* really present in the dataset, and behaviors like the one depicted in Fig.1 might not be correctly classified, simply due to to lack of information about it. In fact, in this case, it is possible that the *jump* of the IP-ID counter occurs later in the sequence, so all the features are evaluated on a resembling simple counter. What practically happens in reality confirms the expectations: about half of the *oddities* are spread between the *global* and *local* implementations. What is instead more surprising is the substantial decrease for the population *random* class. This might be due again to the lack of fundamental information to correctly classify those behaviors. Conversely, constant behaviors are easy to be identified even with a bunch of few packets. These results show that to correctly detect *random* and *odd* IP-ID classes more care is needed: more data might be required to spot the proper behavior of the series. Whilst, for the other classes, few packets are more than enough to correctly classify them.

C. Odd behavior analysis

During the manual labelling phase we have discovered some IP addresses setting IP-ID in unusual manners, which may be ascribable to different reasons, and that we named with the *odd* term. In this section we try to investigate a bit more the *odd* class, making an effort to try to figure out which are the odd behaviors and we try to understand whether we can re-map some of those IP addresses in other classes or not. The first analysis we perform consist in converting the interpretation of the bytes contained in the IP-ID IPv4 header field to little endian and try to perform again the classification to check if something changes. We focus only on the 172,679 IP addresses in $\mathcal{L}$ previously classified as *odd* and perform byte swapping to each IP-ID value of the x, y series. Then, we re-build the dataset with the new features and operate the classifier trained on $\mathcal{G}$ on it. Results show that no meaningful change has occurred, since, execpt for a discardable amount of IP addresses becoming *global*, almost all the IP-ID series remained *odd*.

D. Spatial Analysis

A functional way to graphically visualize the IP-ID classes distribution is through a 12th order Hilbert curve, a fractal space-filling curve which allows the mapping of the one-dimensional IPv4 address space into a bi-dimensional image. The use of Hilbert curves to compactly represent Internet-wide characteristics was first popularized by the Xkcd comic [1] and then used ever since. Each pixel in the image depicted in Fig.16 represents a single /24 prefix block and its color can range among six different hues. Five of these refer to the five IP-ID classes and are respectively assigned to the pixel if one representative address of that /24 network is part of our analysis, i.e. it belongs to $\mathcal{L}$, and the model has

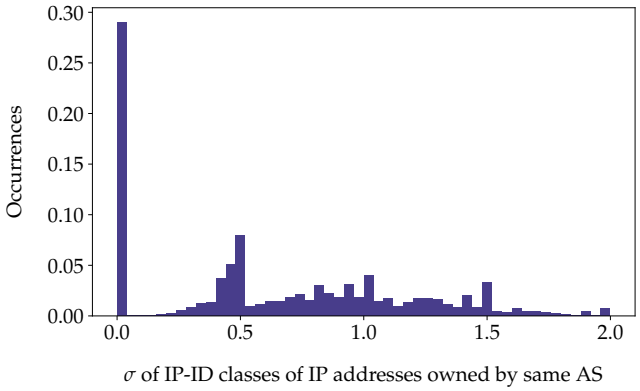

Fig. 14. Standard Deviation of IP-ID classes of IP addresses owned by same AS

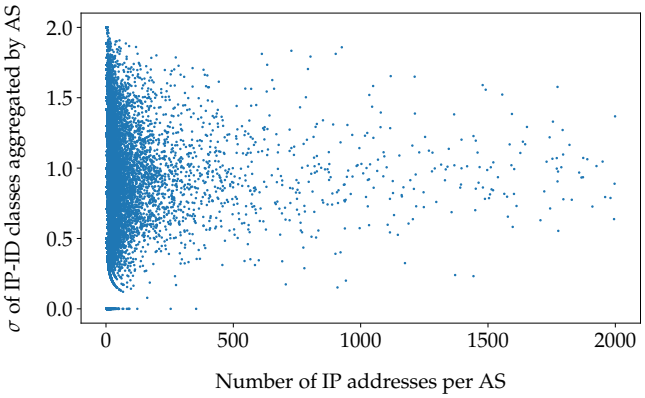

Fig. 15. Standard Deviation of IP-ID classes of IP addresses owned by same AS

classified it as the corresponding color label. On the contrary if there are no IP addresses in $\mathcal{L}$ belonging to that /24 the associated pixel is coloured white. From the image it is clearly possible to highlight some easily distinguishable *islands* of close IP addresses which implement the IP-ID in the same way. However, this is not an exhaustive result to assess that the hosts whose IP addresses belong to the same prefix block generates IP-ID in the same manner.

What can be further inspected is the spatial aggregation of the IP addresses per Autonomous System. We perform this by querying Team Cymru whois database [7] and collecting from there information about the 49189 ASes of the the IP addresses present in our dataset $\mathcal{L}$. We focus only on the 32994 ASes owning at least two IP addresses of the list, discarding in this way 16k IP addresses. We evaluate then the standard deviation σ of the IP-ID classes of the IP addresses belonging to the same AS. We find out, as shown in Fig.14, that 29% of the ASes own IP addresses from whom we collected packets containing the IP-ID generated in the same way, leading to a standard deviation $\sigma = 0$. This result is not telling much if considered alone, and since the most popular class is about 40% of the total this could just be equal to a random clustering of the IP addresses. To have a clearer picture of what is really going on with the AS aggregation, we shrink the data slightly more and focus only on those ASes owning no more than 2000 IP addresses, discarding only 0.5% of the data. We now want to observe how the standard deviation of the IP-ID classes per AS vary with the size of the AS, understood as the number of IP addresses aggregated within the same bin. The scatterplot of this relationship is shown at Fig.15: there we can observe that most of the plot is sparse only in one dimension, i. e. the size of the AS bin. In fact, most of the points lie in the very left region of the graph, in the area where the number of IP addresses per AS is lower than 300.

VI. Conclusions

This work presents, to the best of our knowledge, the first systematic study of the prevalence of different IP-ID behaviors in the current IPv4 Internet (extending this work to IPv6 is a future, necessary, work). In this work, we find evidence that local and constant implementations of the IP-ID are prevalent:

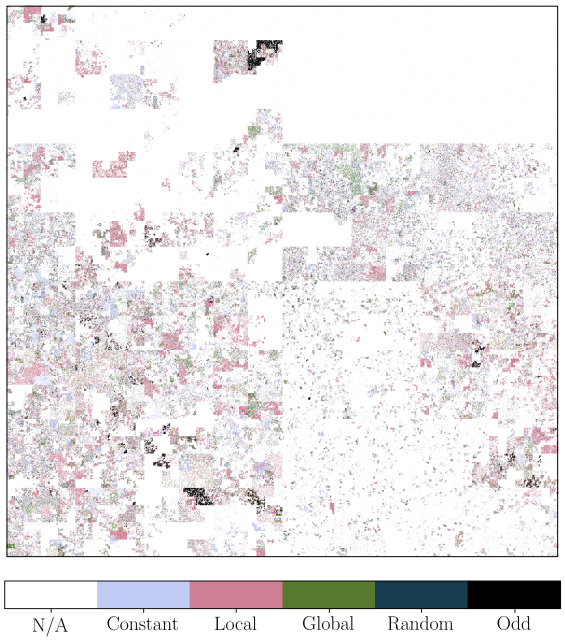

Fig. 16. IP-ID census results, shown as a 12th order Hilbert curve, a fractal space-filling curve that allows the mapping of the one-dimensional IPv4 address space into a bi-dimensional image.

this is in contrast with common knowledge [6], [9], [14], [21], [23], [31], from which the global counter was expected, even in recent times, to be the most popular IP-ID implementation.

Summary and Perspectives. In this study, we first propose a framework to robustly classify the different IP-ID behaviours with only a handful of IP packets. Our methodology consists of three main blocks: data collection, model construction and validation.

The data collection relies on an experimental testbed comprising one sender and two receivers, which collect the IP packets and, specifically, the information related to the IP-ID field. The sender sends a burst of packets, minimizing the impact of external traffic and purposely exploiting spoofing to precisely alternate addresses in the sequence.

In the model construction block, we exploit a decision tree classifier, which is trained and validated over datasets gathered from real measurements and additionally tested in the presence of controlled losses to assess its robustness. Training of the model required manual validation of thousands of sequences: during this phase, we also discovered some odd behaviour, not documented in any of the previous RFCs, and which may be attributed to different reasons.

In instances where odd behaviour was previously reported, our classifier is the first to automatically and correctly label such instances, making it easier to perform large-scale analysis over the Internet. Moreover, classification only requires a handful of packets, making the methodology extremely lightweight.

Given that our classifier is lightweight and robust to losses, we finally perform a census (in 2017) of the IPv4 address space, selecting responsive representatives for each /24 block.

Experimental results show that the majority of hosts adopt local IP-IDs (39%) or a constant counter (34%) of which:

- A fraction of global counters (18%) is significantly lower than expected;
- A non-marginal number of hosts have an odd behaviour (7%);
- Random IP-IDs are only slightly more than an exception (2%).

This outcome provides a picture of Internet-wide adoption of the different IP-ID implementations. Indeed, we gather that the 18% breakdown of the global implementation in 2017 is three times lower with respect to the 57% reported in 2013 [14]. While the quantitative reduction is in line with the statistics reported by recent work that leverages global IP-ID behaviour to detect censorship in the Internet [24], one could have expected the decrease in global implementation to be compensated by an increase of random IP-IDs, which is not the case.

Contributions. Our first contribution is to devise an accurate, lightweight and robust classifier: accuracy of the classifier follows from a principled definition of the statistical features used to succinctly describe the IP-ID sequence; robustness is a consequence of this choice, as features remains wide apart even under heavy losses.

Our second contribution is to carry on a manual investigation effort for a moderate size dataset coming from real Internet measurements: this valuable ground truth allow us to adopt a supervised classification techniques to train a model able not only to detect well-defined behaviors, but also to correctly recognize a wide range of odd behaviors.

Finally, all our datasets, including the testing with manual ground truth, as well as the results of our census, are publicly available at [26]: we hope that the former can assist scientists to build and test new techniques for IP-ID classification, whereas the latter provides practitioners with readily usable lists of the hosts with global IP-ID implementations for their inference. Specifically, the available readily usable list of the approximate half million hosts with global IP-ID implementations global implementations [26] can make work such as [2], [6], [24], [29] still possible. Moreover, by updating and consolidating the scattered knowledge [6], [10], [21], [24], [31] of IP-ID prevalence, this work contributes in refining the current global Internet map.

ACKNOWLEDGMENTS

This work has been carried out at LINCS (http://www.lincs.fr) and benefited from support of NewNet@Paris, Cisco Chair "NETWORKS FOR THE FUTURE" at Telecom ParisTech (http://newnet.telecom-paristech.fr).

REFERENCES

[1] https://xkcd.com/195/.
[2] S. M. Bellovin. A technique for counting NATted hosts. In *Proc. IMW*, 2002.
[3] A. Bender, R. Sherwood, and N. Spring. Fixing ally's growing pains with velocity modeling. In *Proc. ACM IMC*, 2008.
[4] R. Beverly, M. Luckie, L. Mosley, and K. Claffy. Measuring and characterizing IPv6 router availability. In *PAM*, 2015.
[5] R. Braden. *RFC 1122, Requirements for Internet Hosts – Communication Layers*, 1989.
[6] W. Chen, Y. Huang, B. Ribeiro, et al. Exploiting the IPID field to infer network path and end-system characteristics. In *Proc. PAM*, 2005.
[7] T. Cymru. Ip to asn mapping. http://www.team-cymru.org/IP-ASN-mapping.html.
[8] A. Dainotti, K. Benson, A. King, B. Huffaker, E. Glatz, X. Dimitropoulos, P. Richter, A. Finamore, and A. C. Snoeren. Lost in space: Improving inference of IPv4 address space utilization. *IEEE JSAC*, 2016.
[9] K. S. G. Pelletier. *RFC 5225, RObust Header Compression Version 2 (ROHCv2):Profiles for RTP, UDP, IP, ESP and UDP-Lite* , 2008.
[10] Y. Gilad and A. Herzberg. Fragmentation considered vulnerable. *ACM TISSEC*, 2013.
[11] F. Gont. *RFC 6274, Security Assessment of the Internet Protocol Version 4*, 2011.
[12] F. Gont. *RFC 7739, Security Implications of Predictable Fragment Identification Values*, 2016.
[13] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister. Census and survey of the visible internet. In *Proc. ACM IMC*, 2008.
[14] A. Herzberg and H. Shulman. Fragmentation considered poisonous, or: One-domain-to-rule-them-all. org. In *IEEE CCNS*, 2013.
[15] Idle scanning and related IPID games. https://nmap.org/book/idlescan.html.
[16] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Measurement and classification of out-of-sequence packets in a tier-1 IP backbone. *IEEE/ACM TON*, 2007.
[17] K. Keys, Y. Hyun, M. Luckie, and K. Claffy. Internet-scale IPv4 alias resolution with MIDAR. *IEEE/ACM TON*, 2013.
[18] A. Klein. OpenBSD DNS cache poisoning and multiple O/S predictable IP ID vulnerability. Technical report, 2007.
[19] W.-Y. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2011.
[20] M. Luckie, R. Beverly, W. Brinkmeyer, et al. Speedtrap: internet-scale IPv6 alias resolution. In *Proc. ACM IMC*, 2013.
[21] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level internet path diagnosis. *ACM SIGOPS Operating Systems Review*, 2003.
[22] J. C. Mogul and S. E. Deering. *RFC 1191, Path MTU discovery*, 1990.
[23] S. Mongkolluksamee, K. Fukuda, and P. Pongpaibool. Counting NATted hosts by observing TCP/IP field behaviors. In *Proc. IEEE ICC*, 2012.
[24] P. Pearce, R. Ensafi, F. Li, N. Feamster, and V. Paxson. Augur: Internet-wide detection of connectivity disruptions. In *IEEE SP*, 2017.
[25] J. Postel. *RFC 791, Internet protocol*, 1981.
[26] F. Salutari, D. Cicalese, and D. Rossi. https://perso.telecom-paristech.fr/drossi/dataset/IP-ID/.
[27] F. Salutari, D. Cicalese, and D. Rossi. A closer look at ip-id behavior in the wild. In *International Conference on Passive and Active Network Measurement (PAM)*, Berlin, Germany, Mar 2018.
[28] F. Salutari, D. Cicalese, and D. Rossi. A closer look at ip-id behavior in the wild (extended tech. rep.). Technical report, Telecom ParisTech, 2018.
[29] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with rocketfuel. *IEEE/ACM TON*, 2004.
[30] J. Touch. *RFC 6864, Updated Specification of the IPv4 ID Field*, 2013.
[31] M. A. West and S. McCann. *RFC 4413, TCP/IP field behavior*, 2006.

[32] S. Zander, L. L. Andrew, and G. Armitage. Capturing ghosts: Predicting the used IPv4 space by inferring unobserved addresses. In *Proc. ACM IMC*, 2014.