On the Impact of Sampling on Traffic Monitoring and Analysis

Antonio Pescapé[1], Dario Rossi[2], Davide Tammaro[1,2,⋆], Silvio Valenti[2]

[1] Università di Napoli Federico II, Italy – pescape@unina.it
[2] Telecom ParisTech, France – first.last@enst.fr

Abstract. Motivated by significant advances in transmission technology, with the corresponding increase of link rates, *traffic sampling* is becoming a normal way of operation in traffic monitoring. Given this trend, our aim is to assess the impact that sampling may have on a wide range of tasks which are typical of an operational network. We follow an experimental approach, exploiting passive analysis of network traffic flows, taking into account different sampling policies (e.g., systematic, uniform and stratified) and sampling rates. We consider a large set of "*features*" (i.e., any kind of properties characterizing the traffic flows, from packet size and inter-arrival time, to Round Trip Time, TCP congestion window size, number of out-of-order packets, etc.) which are representative for a rather wide class of applications, such as traffic monitoring, analysis, accounting, classification, etc. We then quantify the amount of degradation and bias that sampling introduces with respect to the unsampled traffic by means of several statistical measures. From our experimental campaign, carried on over several real traffic traces representative of different operational networks, we observe that a significant degradation affects a wide number of features. At the same time, we also find that the set of features that are least degraded is consistent across datasets. Our findings also partly counter earlier observations concerning the better estimation quality achievable under stratified sampling: considering a wide set of features, no significant reduction of the estimation bias can be obtained by merely tweaking the sampling policy.

1 Introduction and Motivations

Due to ever growing line speed and Internet traffic amount, measurement of network traffic generates a massive volume of data introducing scalability issues in both storage and processing. Two orthogonal approaches can be used to reduce this amount of data – namely, *aggregation* and *sampling*. Although *data aggregation* is a core technique in current Internet, as the widespread use of SNMP (Simple Network Management Protocol) testifies, nevertheless there are many operations (such as billing, management, SLA compliance verification, classification, etc.) that require information pertaining to individual flows, rather than to flow aggregates. As such, *sampling* has become an integral part of passive network measurements, and much work has already been done in this field [2, 8, 7, 11, 15, 10, 6, 27, 12, 21, 18, 3, 17, 16]. Several studies focus on the design of sampling policies and on their impact, typically considering a few metrics

[⋆] This work has been carried out during the internship of Davide Tammaro at Telecom ParisTech.

only [7, 10, 21, 17, 25, 5], whereas other works concentrate on a single application of sampling [15, 27, 16, 18, 3, 22].

In this paper, we aim at considering a larger number of metrics and at quantifying their robustness under sampling. We believe that this wider perspective can be instrumental to a number of applications (e.g., monitoring, classification, anomaly detection, etc.), without being bound to a single one in particular. The robustness of the metrics is evaluated in terms of statistical indexes, such as the *Bhattacharya distance*, the *Hellinger Distance*, and the *Kullback-Leibner divergence* of sampled versus unsampled data. The proposed methodology is based on a popular flow-level analyzer [1] which operates on packet level traffic producing a wealth of statistical features under different sampling policies. By performing offline analysis of passive traces, we are able to compare the results gathered from sampled traffic with the corresponding results of unsampled traffic, so to assess the extent of the degradation of the considered features. As a side effect, the analysis of the introduced distortion will also give insights on the amount of coding work required to instrument traffic monitoring software to cope with sampling even for non-trivial metrics (e.g., TCP cwnd or anomalous conditions).

Obtained results - carried on several real traffic traces, some of which are publicly available - show a qualitative agreement across multiple datasets: the set of features which proved robust to sampling strategies is the same over different datasets, although the magnitudo of the degradation may vary. Metrics whose reliable estimation depends on several packets are clearly more degraded, and ad-hoc sampling techniques may be required to reduce the sampling impact. Not only this translates into greater research effort, but it might also lead to scenarios where several sampling processes would run in parallel, as each measure requires a special sampling technique. Finally, as early work [22, 16] has already shown, the degradation of a metric introduced by sampling does not necessarily reflect in an equal reduction of performance of successive applications (e.g., anomaly detection, traffic classification) operating on that measure. Thus, a mild amelioration of the estimation quality may be enough to allow a useful exploitation of sampled data. In future work we aim at exploring this trade-off.

The reminder of this paper is organized as follows. In Sec. 2 we overview the most relevant work, highlighting the relations with our study. We describe the followed methodology in Sec. 3, detailing the tools used, the dataset to which we apply them, and the metrics that we use for the quantitative assessment. Results of our experimental campaign are reported in Sec. 4. Finally conclusive remarks and future directions are discussed in Sec. 5.

2 Related Work

Due to the crucial role of packet sampling, several works have already been published on this topic. While it is out of scope to provide a throughout survey of these studies, for which we refer the reader to [9], we nevertheless need to better position our paper with respect to that work.

In [2] researchers have started agreeing on a categorization of packet sampling techniques, which has since then evolved until recently becoming an IETF standard document [29]. Basically, sampling techniques can be categorized depending on the selection scheme, which can be *deterministic* (or systematic), *random* or possibly *content-dependent*, with some further subcategories exhaustively presented in [29]. Moreover, the selection trigger can be either based on the amount of *time* elapsed or on the number of *packets* transmitted between two consecutive samples. Initially, researchers investigated and compared different random selection schemes (possibly including stratification) and triggers [7], proposing then more sophisticated techniques based on hash functions [11], sample and hold [12], and hash-based sketches [17]. Other works focused instead on making the sampling rate *adaptive* [6, 8, 15], for instance to the traffic load.

Major results can be summarized from the above works. First, authors of [7] showed that sampling triggers based on the count of packets are more robust than time-based triggers, which cope badly with the bursty nature of data traffic. They also point out the advantages of random sampling, due both to its intrinsic statistical robustness and to its higher resilience to evasion/attacks. The inherent robustness of *random sampling* (and especially of stratified sampling [9]) has been also pointed out in [7, 23], although more recent results [5] suggest that the statistical multiplexing of traffic can have the same effect of a random selection process. In fact, [5] shows that volume information (e.g., packets, bytes) obtained through deterministic 1-out-of-k packet sampling is equivalent to random packet sampling with rate $p = 1/k$.

Researcher have also highlighted that specific sampling techniques may be more effective for different tasks or features – such as trajectory sampling for spatial properties [11], sketches for [17] flow-size and so on. Moreover, most work to date focuses on specific metrics, essentially accounting for traffic volumes under sampling [10, 21, 17, 25, 5]. More recently researchers have started investigating the impact that sampling may have on a wider range of applications, such as network management [15], SLA verification [27], traffic classification [16, 4] or anomaly detection [18, 3, 22]. This shift in the application focus also implies a shift on the quantities that have to be measured – e.g., from simple volumes of traffic [10, 21, 17, 25, 5] to other properties, or "features". However these works consider the effect of sampling only on the performance of a specific application (e.g., precision and recall of anomaly detection or traffic classification, SLA compliance). While this is a very useful effort, nevertheless results may be bound to the specific technique used for that task, thus measuring the joint effect of sampling on the metrics and on the discriminative power of the considered underlying machine learning (ML) tool.

In this work we adopt a complementary approach, focusing on the impact of sampling on the measure of relevant traffic *features*, irrespectively of their actual usage. Under this light, [13] is a work closer to ours, even if not directly related to sampling, as it investigates the relative stability of different metrics across different datasets (although [13] focuses again on a specific application, namely traffic classification). In [4] another closer contribution to ours is proposed: mainly, obtained results indicate that the accuracy of standard ML tools degrades drastically with sampling. In our work, by considering different features over different traces, we quantify instead the amount of "distortion" that different sampling policies and rates introduce on the measurement process.

3 Methodology

We first define the *sampling policies* (Sec. 3.1) that we take into account. We then briefly describe the *datasets* (Sec. 3.2) used throughout this work, and afterwards we elaborate on the list of *features* (Sec. 3.1) that we focus on. Finally, we introduce the different statistical *metrics* (Sec. 3.4) used to evaluate the distortion induced by sampling, further using a few features as preliminary examples (Sec. 3.5).

3.1 Sampling Policies

We implement different sampling policies as defined [29]. For the time being, we have implemented "unbiased" sampling techniques, leaving biased techniques as a future work; we however shed preliminary insights on this issues in Sec. 5. In more details, we consider:

- **Systematic sampling**: packets are sampled in a deterministic fashion, with 1-out-of-k packets selected;
- **Random sampling**: packets are sampled at random, each packet is sampled independently at a rate $p = 1/K$;
- **Stratified sampling**: packets are sampled at random, but during a window of k consecutive packets only 1 packet can be sampled.

3.2 Dataset

In order to gather results that are representative of a wide range of network environments and epochs, we use several traces, whose main features are summarized in Tab. 1. Namely, the table reports the capture year and the number of packets, flows and different IP hosts observed in the traces. In more details, the traces refer to:

- **Campus** is a 2-hours long trace captured during 2008 from our network, representive of a typical data connection to the Internet. LAN users can be administrative, faculty members and students. Most of the traffic is due to TCP data flows carrying Web, email and bulk traffic, since a firewall blocks all P2P file sharing applications.
- **ISP** is a 1-hour long trace collected during 2006 from one of the major European ISP, which we cannot cite due to NDA, offering triple-play services over broadband access. ISP is representative of very heterogeneous scenario, in which no traffic restriction applies to customers.
- **Auckland-VI** [14] is continuous 4 and a half day GPS-synchronized IP header trace captured by means of a DAG card. It contains the totality of traffic flowing through the University of Auckland Internet infrastructure and was collected during the year 2001, on the router that connected the ISP ATM switch to a 100 Mbps Ethernet hub.

Table 1. Summary of dataset used in this work.

Trace	Year	Packets	Flows	IPs
ISP	2006	44,396,297	219,481	61,959
Campus	2008	17,246,459	422,928	81,687
Auckland-VI	2001	291,052,998	11,128,910	410,059

3.3 Features

Tstat [1] logs several traffic features, which are in part per-flow metrics, and in part aggregated indexes. Moreover, for certain properties Tstat is able to distinguish the traffic directionality (e.g., incoming versus outgoing versus local, and client-2-server versus server-2-client) of the measurement. A summary of such properties is reported in Tab. 2, divided according (i) the corresponding layer as well as (ii) the number of packets needed to perform the measure, as some features can be directly derived from a single packet (e.g., packet length), while others require multiple packets to be evaluated (e.g., packet inter-arrival). It is important to notice that there is a good match with the about 240 features listed in [20], which contains the most relevant features for traffic classification. Yet, we point out that our work uses these features with a different semantic from [20], as we will be expressing the feature distortion mostly in its *aggregated* form, whereas traffic classification needs measures at an *individual* flow level–an interesting aspect that we leave for future work.

3.4 Metrics

In order to quantify the distortion introduced by the sampling procedures, we consider different statistical indexes. Denote by P an unsampled feature, which is described by the probability density function $p(x)$ measured over the traffic aggregate. Denote by Q the same feature as measured under a sampling process, which is then described by the probability density function $q(x)$ measured over the sampled traffic. To express the distance between $p(x)$ and $q(x)$ we consider the following standard metrics:

– **Kullback-Leibner (KL)**

$$KL(p\|q) = \sum_{x\in X} p(x) \ln \frac{p(x)}{q(x)} \tag{1}$$

– **Bhattacharyya Distance (BD)**

$$BD(p,q) = -\ln \sum_{x\in X} \sqrt{p(x)q(x)} \tag{2}$$

– **Hellinger Distance (HD)**

$$HD(p,q) = \sqrt{1 - \sum_{x\in X} \sqrt{p(x)q(x)}} \tag{3}$$

Table 2. List of features considered. Single star sign (⋆) denotes features considering incoming vs. outgoing vs. local directions. Double star sign (⋆⋆) denotes features considering also client-2-server vs. server-2-client directions (in addition to incoming vs. outgoing vs. local directions).

IP layer	Single IP datagram	UDP layer	Single UDP segment
ip_tos⋆	TOS field	udp_port_flow_dst	Destination port per flow
ip_ttl⋆	TTL field	udp_port_dst⋆	Destination port per segment
ip_len⋆	Packet length [byte]	udp_tot_time	Flow lifetime [ms]
ip_bitrate⋆	Bitrate [kbit/s]	udp_cl_b_l⋆	Flow length [byte], coarse granularity
ip_protocol⋆	Protocol type	udp_cl_b_s⋆	Flow length [byte], fine granularity
addresses	internal variable	udp_cl_p⋆	Flow length [packet]

TCP layer	Single TCP segment	RTCP layer	Multiple RTCP segments
tcp_mss_used	Negotiated MSS	rtcp_bt⋆	Average bitrate [bit/s]
tcp_mss_b	MSS declared by Server	rtcp_mm_bt⋆	Associated MM flow bitrate[kbit/s]
tcp_mss_a	MSS declared by Client	rtcp_mm_cl_b⋆	Associated MM flow length [bytes]
tcp_opts_TS	Timestamp option	rtcp_mm_cl_p⋆	Associated MM flow length [packets]
tcp_opts_WS	WindowScale option	rtcp_t_lost⋆	Lost packets per flow
tcp_opts_SACK	SACK option	rtcp_f_lost⋆	Prob. of lost packets
tcp_bitrate⋆	Application bitrate	rtcp_dup⋆	Duplicated packets
tcp_port_syndst⋆	Destination port (SYN segments only)	rtcp_lost⋆	Lost packets
tcp_port_synsrc⋆	Source port (SYN segments only)	rtcp_jitter⋆	Average jitter
tcp_port_dst⋆	Destination port (all segments)	rtcp_rtt⋆	RTCP Round trip time [ms]
tcp_port_src⋆	Source port (all segments)	rtcp_avg_inter⋆	Average inter-packet gap (IPG)
		rtcp_cl_b⋆	RTCP flow length [bytes]
		rtcp_cl_p⋆	RTCP flow length [packets]

TCP layer	Multiple TCP segments	Multimedia (MM)	Multiple MM packets
tcp_interrupted	Early interrupted flows[26]	mm_burst_loss⋆	Burst length of lost packets [packet]
tcp_thru ⋆⋆	Application throughput [Kbps]	mm_p_late⋆	Prob. of late packets
tcp_tot_time	Flow lifetime	mm_p_lost⋆	Prob. of lost packets
tcp_rtt_cnt	RTT: number of samples	mm_p_dup⋆	Prob. of duplicate packets
tcp_rtt_stdev	RTT: standard deviation [ms]	mm_p_oos⋆	Prob. of out-of-sequence packets
tcp_rtt_max	RTT: maximum RTT [ms]	mm_n_oos⋆	Length of out-of-sequence burst
tcp_rtt_avg	RTT: average RTT [ms]	mm_oos_p⋆	Total out-of-sequence packets
tcp_rtt_min	RTT: minimum RTT [ms]	mm_reord_p_n⋆	Total reordered packets
tcp_cl_b_l	Flow length [byte], coarse granularity	mm_reord_delay⋆	Delay of reordered packets
tcp_cl_b_s	Flow length [byte], fine granularity	mm_avg_jitter⋆	Average jitter [ms]
tcp_cl_p	Flow length [packet]	mm_avg_ipg⋆	Average IPG [ms]
tcp_cwnd	TCP in-flight-size [byte]	mm_avg_bitrate⋆	Stream bitrate [kbit/s]
tcp_win_max	TCP max RWND [byte]	mm_cl_b⋆	Long stream flow length [bytes]
tcp_win_avg	TCP average RWND [byte]	mm_cl_p⋆	Long stream flow length [packet]
tcp_win_ini	TCP initial RWND [byte]	mm_cl_b_s⋆	Short stream flow length [bytes]
tcp_anomalies	TCP anomaly [19]	mm_cl_p_s⋆	Short stream flow length [packet]
	⋆⋆: tcp_rtx_RTO , tcp_rtx_FR,	mm_tot_time_s⋆	Short stream flow lifetime [ms]
	tcp_flow_ctrl, tcp_net_dup,	mm_tot_time⋆	Stream flow lifetime [s]
	tcp_unnrtx_FR , tcp_unnrtx_RTO,	mm_rtp_pt⋆	RTP payload type
	tcp_reordering, tcp_unknown	mm_uni_multi⋆	Unicast/multicast flows
		mm_type⋆	Stream type

KL is used in [24] to reduce the data set size, in an approach complementary to sampling. BD and HD are instead typically used as a score of similarity between metrics, and have been used in [28] to assist the context of classification as well.

Besides these statistical indexes, we also take into account some other metrics:

- **Packet-Accuracy (PA)**: percentage of packets selected by the sampling procedure; this is known by design;
- **Flow-Accuracy (FA)**: percentage of flows whose packets are selected by sampling;

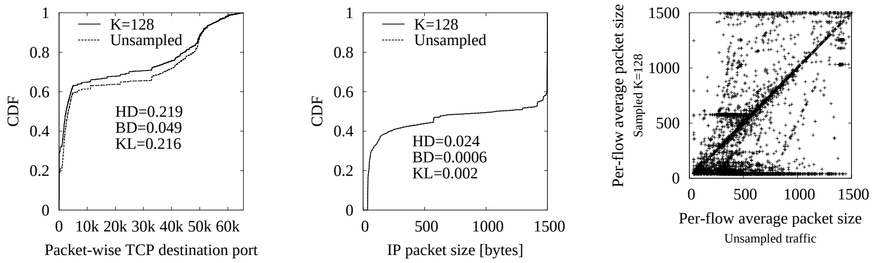


Fig. 1. CDF of two example features, namely (a) number of packets per destination TCP port and (b) IP packet size. Plots report the CDF gathered from the unsampled vs sampled traffic aggregate, along with the statistical indexes of distortion. At the individual-flow level, (c) reports instead the scatter plot of the unsampled versus sampled average per-flow IP packet size.

- **Byte-Accuracy (BA)**: percentage of bytes carried by flows which are selected by sampling; note that this metric takes into account all packets of those flows of which at least one packet has been sampled.

These accuracy measures allow us to understand the actual amount of data on which the other metrics are evaluated. In particular with our proposed BA definition we want to asses the real amount of bytes of the total unsampled traffic corresponding to the sampled flows. Notice also that the amount of bytes actually sampled is roughly proportional to the packet accuracy metric, while our BA accuracy expresses a diverse metrics of the sampled traffic.

3.5 Examples

For the sake of clarity, let us make a preliminary exploratory example of the selected metrics and features, so to have a first idea of the scale of the distortion scores defined so far. Fig. 1-(a) and Fig. 1-(b) report the CDF of two features, respectively counting (a) the number of packets directed to a given TCP port and (b) the IP packet size in bytes. CDFs are reported for both original unsampled traffic, as well as for uniformly sampled traffic with $k = 128$. Values of different distortion metrics are reported in the picture. The CDF of the packet-wise destination port Fig. 1-(a) shows a moderate distortion, with a corresponding degradation slightly above $HD = 0.2$: in this case, differences in the CDF, although modest, can be seen with naked-eyes from the plot. Conversely, packet size shows a degradation score of about one order of magnitude smaller $HD = 0.025$: in this case, no remarkable difference appears from the plot.

To better understand the distortion score, let us dig further in this example, considering the more robust feature, i.e., IP packet length. As previously stated, some tasks (e.g., monitoring, accounting, etc.) need to consider features at a traffic aggregate level, whereas other tasks (e.g., traffic classification, QoS management, etc.) rather have to consider features at an individual flow level. While we leave a thorough analysis of this second viewpoint for future work, this examples gives us some preliminary insights

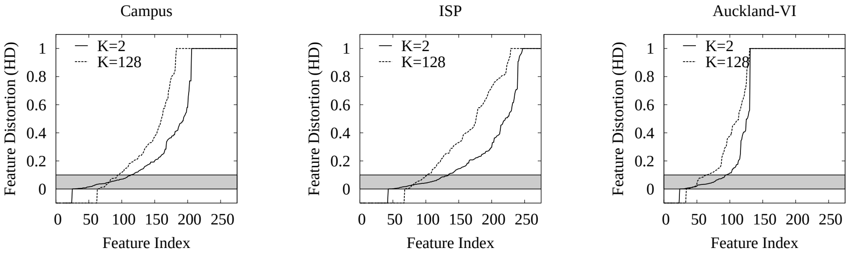


Fig. 2. Features distortion for all datasets in terms of HD distance for two sampling granularities $k \in \{2, 128\}$.

on the relationship between the two viewpoints. Fig. 1-(c) shows a scatter plot of the sampled versus unsampled metrics whose CDF is shown in Fig. 1-(b). More in details, the x-axis represent the per-flow average IP packet size considering unsampled traffic, while the y-axis shows the same metric measured on sampled traffic.

Notice that, while many points align over the $y = x$ line, indicating good correlation between sampled and unsampled data even at flow-level, we can notice a number of points falling in a few horizontal lines (namely $y = 40, 576, 1500$). We found that for these flows only a packet was sampled, which is not representative of the average packet size. In fact with a single sample, it is likely to get a typical-sized packet (e.g, a $40-$bytes packet without data, or $1500-$bytes full payload packet, or a $576-$bytes packet) which will lead to a bad estimation of the actual average packet size of the flow (which is represented on the x-axis). In this case, other metrics may better represent the distortion of the sampled population (e.g., such as the correlation coefficient, the relative error, the root mean square error, etc.), which we aim at investigating in future work.

4 Experimental Results

In this section, we first report a preliminary evaluation of sampling impact by considering *all features* but focusing on a smaller set of sample rates, policies and metrics (Sec. 4.1). We then concentrate on a *reduced features set*, which comprises the most robust features under sampling, over which we conduct a thorough sensitivity analysis across a wider range of sampling policies, rates and metrics (Sec. 4.2).

4.1 Features distortion

To build an overall picture of the sampling impact on the whole set of features we start by considering systematic sampling and a sampling rate $1/k$ with $k \in \{2, 128\}$. For each feature, we calculate the CDF for both unsampled and sampled traffic, and compare the two distributions by means of their HD score. Notice that, although for the time being we limitedly consider only the HD metric, results gathered in this section

Table 3. Breakdown of robust features per protocol layer and dataset.

		Number of robust features ($HD \leq 0.1$)											
Protocol	Features	K=2						K=128					
Layer	Number	Campus		ISP		Auck-VI		Campus		ISP		Auck-VI	
IP	15	12	80%	14	93%	9	60%	10	66%	12	80%	8	53%
TCP	121	58	48%	45	37%	54	44%	17	14%	8	7%	12	9%
UDP	17	9	53%	9	53%	8	47%	3	18%	5	29%	4	23%
RTCP	39	4	10%	13	33%	-	-	-	-	4	10%	-	-
MM	63	4	6%	9	14%	-	-	-	-	1	1%	-	-
All	255	87	34%	90	35%	71	28%	30	12%	30	12%	24	9%
	$\cap$	50/255 (20%)						15/255 (6%)					
	$\cup$	128/255 (50%)						52/255 (20%)					

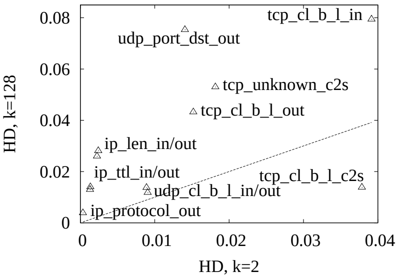
hold even when considering other similarity scores (even though the actual range of values assumed varies across metrics).

In Fig. 2, we report the feature index on the x-axis, and on the y-axis the corresponding value of the HD for that feature. The features order actually differs from trace to trace, as they are sorted for increasing distance so that the represented curves are monotone. We remind that $HD = 0$ means that two distributions are identical, while $HD = 1$ means that two distributions are completely different. Features with $HD < 0$ are those whose HD values could not be computed (e.g., due to missing samples).

First, we notice that the overall impact of sampling is heavily dependent on the dataset. For instance, we see that in the Auckland trace roughly half of the features are completely distorted by sampling (i.e., $HD = 1$) at both high and low sampling rate. On the contrary, the ISP dataset seems more robust to sampling, as features exhibit a lower degradation. Moreover the distortion appears smoother, with many features scoring a medium HD value, while in the Auckland trace, transition from small to large HD scores appears instead sharper. We attribute this behavior to the different traffic mixture of the traces: as the ISP traces contain a much more various mixture of traffic, sampling impact "spreads" over a larger amount of features. Finally, the Campus trace shows a behavior which is in the middle of this two extremes. Second, as expected the sample rate has an important effect, with *almost* all features showing a degradation when we increment the sampling step from $k = 2$ to $k = 128$. Actually, for some features we assist to the opposite, counter-intuitive behavior, on which we will return later in this section.

We now aim at selecting a small set of features that are robust to sampling, on which to perform a more thorough sensitivity analysis. We thus define a threshold, that we arbitrarily set to $HD = 0.1$, to individuate the features which are robust to sampling: we consider features whose HD value falls below the threshold to be robust. Notice that the selected HD threshold is about half of the distortion early shown in Fig. 1-(a), where discrepancy of the CDFs was clearly visible although not massive.

The gray strips in Fig. 2 helps in visually identifying the amount of robust features. In Tab. 3, we report the number of such features, as well as the percentage over the total number of features. In order to better identify the most robust ones, the table reports their breakdown with respect to the protocol layer partitioning, earlier introduced in



	$k = 2$		$k = 128$	
Trace	FA	BA	FA	BA
Campus	0.938	0.934	0.138	0.727
ISP	1.125	0.992	0.197	0.943
Auck-VI	1.130	0.999	0.136	0.687

Fig. 3. (a) Scatter plot of HD distortion for the robust feature subset, with sampling $k = 2$ and $k = 128$ and (b) flow accuracy-FA and byte-accuracy BA values.

Tab. 2. As expected, features that solely require the inspection of packet level headers are more robust than features that need the observation of multiple packets of the same flow. As a consequence, IP-layer features clearly show the highest robustness: particularly, for the ISP traces and even for high sampling granularities, about of 80% of its features fall below the threshold. On the other hand, many features at the TCP/UDP-layer need to be measured over multiple packets, consequently only very few of them appear robust under sampling. Already with $k = 2$, more than half of them exceed the threshold, even for the ISP dataset, which represents our best-case. Furthermore, when considering multimedia MM and RTCP traffic, we can see that robustness is further compromised by another factor: the relatively low amount of MM and RTCP traffic present in the original traces reduces the number of valid samples for these features. Notice moreover that no MM/RTCP traffic was present in Auckland dataset.

Finally, we quantify to what extent the set of robust features keeps the same across datasets. We consider therefore the intersection $\cap$ and the union $\cup$ of the robust features over all the datasets, and report their size in the last two rows of Tab. 3. Results show that, when considering a small sampling factor $k = 2$, about 50 features are robust to sampling: robust features common to all three datasets account in this case for more than half of the robust features in each dataset (55% in ISP and 70% in Auckland), corresponding to about 20% of the total features computed by Tstat. When considering $k = 128$, about 15 features are common to all three dataset, i.e. more than half of the robust features in each dataset. Thus, we may say that *the prominent portion of features that are robust is also consistent across datasets*. Conversely, observing that the union $\cup$ of the robust features is larger than the intersection $\cap$, we can conclude that the distortion experienced by different features may differ widely from one dataset to another: in other words, it seems as though *some features are robust only under some particular traffic conditions*.

Before delving into the impact of different sampling policies and rates, let us focus more closely on the features that are robust across all datasets. In the scatter plot in Fig. 3-(a), we represent each of these features with a single point, whose coordinates are respectively its HD value for $k = 2$ and $k = 128$. In the picture, we label each point with the name of the corresponding feature. Intuitively, we could expect all the points falling in the upper part of the graph above the $y = x$ bisector, since metrics should deteriorate at higher values of k. Conversely, we find that some features exhibit an opposite and counter-intuitive behavior. For instance, this effect is particularly evident for the `tcp_cl_b_l_c2s` feature, i.e. the TCP flow lenght, measured with a coarse granularity. In this case, for larger sampling steps, many short flows are no longer sampled, with a corresponding decrease of the mass of flows falling into the smallest bin. Thus, in this case the feature estimation ameliorate at higher sampling as joint consequence of the traffic nature (since sampling tends to select multiple packets from the same elephant flows) and of the specific binning adopted.

Finally, in order to understand the quantity of traffic to which our investigation refer to, in Fig. 3-(b) we report the flow-accuracy (FA) and byte-accuracy (BA) for the two sampling granularities for each considered dataset. As it can be seen, at low sampling step $k = 2$, the number of flows articially inflates for the ISP and Auckland traces: as already observed in [10], long flows can be split if the time between sampled packets exceeds the flow timeout (which is by default set to 200 seconds in Tstat), possibly resulting in an over-estimation of the actual number of flows. This is especially visible for $k = 2$, since for $k = 128$ the effect of short flows under-sampling has a greater impact, overall reducing the ratio of seen flows. On the other hand, we observe that the byte accuracy is always very high, meaning that results reported in this paper are representative of the bulk of traffic.

4.2 Sensitivity analysis

In this section, we perform a careful evaluation of the impact of different sampling policies and rates on the estimation accuracy of the features belonging to the robust set defined in the previous section. We report the results of our sensitivity analysis in Fig. 4: graphs are arranged in a matrix, whose columns correspond to the different sampling policies, while the rows are related to the different statistical metrics used to quantify the feature distortion. For each sampling policy, we employed an exponentially increasing sampling step $k = 2^i, i \in [1 \dots 10] \subset \mathbb{N}$, reported on the x-axis of every plot. Each graphs contains three curves, one for each dataset, depicting the average distance score over the 15 features belonging to the robust set. Variance of the distance score is also reported, by means of vertical bars. Notice that even if HD and BD can be derived one from the other, this does not hold for their average values reported in the picture, because the relation between the two distances is not linear.

At first glance we can observe that, as expected, the lower the sampling rate, the larger the feature distortion of the measures: notice however that the mean of the distances is *almost* always increasing, with some exceptions that we will consider later on. Notice also that, if we look at the graph related to the Hellinger distance, the mean of the features remains under the threshold even under heavy sampling: this testifies that, despite an overall degradation, with some features likely exceeding the threshold, the set of features generally keeps rather robust to sampling.

Let us now focus on the different metrics by comparing graphs belonging to the same column. It is easy to gather that, although each metric takes values in different ranges, all of them exhibit a very similar qualitative trend. Furthermore, when considering low sampling factors, the three metrics agree in identifying the Campus dataset

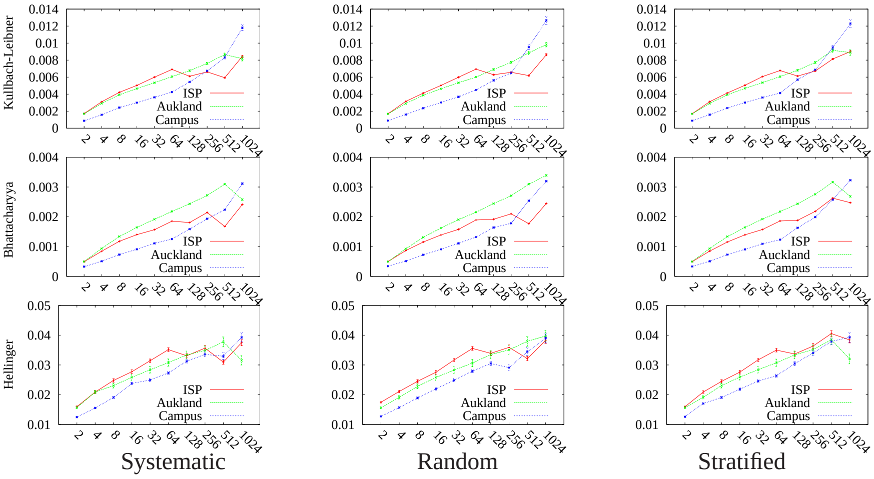


Fig. 4. Mean and variance of the statistical metrics calculated over the robust set of features across different sampling policies and rates.

as the best case with respect to the robust feature set, whereas the ISP and Auckland traces have very similar scores. For high sampling granularities instead, the Campus trace shows the largest degradation in all our experiments. Interestingly, the ISP trace exhibits instead a non-monotonous behavior at heavyer sampling: this is effect is the same already shown in Fig. 3, and is due to features whose estimation improves under sampling, because they are most related to long-lived flows. Notice that this is particularly evident for the ISP trace, which is characterized by a larger number of "elephant" flows, as one could easily infer from Tab. 1.

Finally, comparing graphs on the same row, we can assess the effect of the considered sampling policy. Interestingly, the main observation is that, when a larger number of features are observed (as opposite to a single feature), it seems as though the influence of the sampling policy is very modest. Indeed, differences in the anomaly scores due to sampling policies are practically negligible for low sampling factors. Only for high values of k, stratified sampling exhibits a slightly better behavior, as it partly eliminates the aforementioned non-monotonous behavior. However, systematic and random sampling show almost the same performance: we individuate the cause of such similarity in the statistical multiplexing of network traffic, which makes the difference between systematic and random sampling almost negligible when considering a large set of features. This is a rather interesting finding, that counters the conclusions reached in some previous work [7] and that further extends some similar results of recent works [5], which were however limited to traffic volumes.

5 Conclusions

In this paper, we have investigated the impact of packet sampling on network traffic monitoring and analysis. Aiming at a comprehensive study, we have (i) implemented three different sampling policies with increasing sampling granularities, (ii) considered a vast set of packet-level and flow-level features of network traffic, and (iii) applied our methodology to a fairly large dataset of very heterogeneous traces. By running a modified version of Tstat, a flow-level traffic analyzer, we have been able to compare the results obtained with sampled versus unsampled traffic data. Comparison has been expressed in terms of several statistical indexes, apt at quantifying the amount of feature degradation introduced by sampling.

Our results show that, on the one hand, sampling causes an important degradation of the feature estimation: indeed, most of the features are already severely distorted at low sampling rates. On the other hand, we have found that there exists a small number of features intrinsically robust to sampling, which is furthermore consistent across all the considered datasets. The sensitivity analysis conducted on this reduced set of features further points out that, unlike previous studies have shown, the specific sampling policy employed may only have a minor impact on reducing the feature distortion. We identify two main reasons behind this finding: first, the statistical multiplexing may partly eliminate the bias induced by simple strategies (e.g., systematic sampling) and second, this evidence may have been hidden by previous work which typically focused on a few specific features only (e.g., mainly packet size and traffic volumes). Finally, we have also pin-pointed a number of counter-intuitive behaviors of some features, whose estimation quality improves under heavy sampling, and that shows the importance of correctly assessing the impact of sampling even on simple features.

In future work, we aim at extending this work in several directions. First, we would like to consider a larger set of sampling strategies, such as non-uniform sampling policies (e.g., sample all TCP packets with SYN flag set, sample a batch of consecutive packets, etc.). Indeed, these strategies improve the estimation of some features (e.g., SYN sampling ameliorates flow-length, while batch-sampling is useful for packet interarrival), and their impact on other features is worth investigating as well. Second, we aim at considering a wider range of application (e.g., traffic classification, anomaly detection, etc.) so to better correlate the feature distortion with the performance of the application itself. Finally, another interesting issue consists in making the analyzer tool aware of the packet sampling policy and rate, which could possibly assists a better estimation for some particular features.

References

1. Tstat, `http://tstat.tlc.polito.it`.
2. P.D. Amer and L.N. Cassel. Management of sampled real-time network measurements. In *Proc. of IEEE LCN '89*, Oct 1989.
3. D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina. Impact of packet sampling on anomaly detection metrics. In *Proc. of ACM SIGCOMM IMC '06*, Rio de Janeriro, Brazil, Oct 2006.
4. V. Carela-Espaol, P. Barlet-Ros, and J. Sol-Pareta. Traffic classification with sampled netflow. *Technical Report, UPC-DAC-RR-CBA-2009-6*, Feb. 2009.
5. Y. Chabchoub, C. Fricker, F. Guillemin, and P. Robert. Deterministic versus probabilistic packet sampling in the Internet. In *Managing Traffic Performance in Converged Networks(LNCS)*, Ottawa, Canada, Sep. 07.

6. B. Choi, J. Park, and Z. Zhang. Adaptive random sampling for load change detection. In *Proc. of ACM SIGMETRICS '02*, Marina Del Rey, CA, US, Jun 2002.
7. K. C. Claffy, G. C. Polyzos, and H. Braun. Application of sampling methodologies to network traffic characterization. In *Proc. of ACM SIGCOMM '93*, San Francisco, CA, USA, Sep 1993.
8. J. Drobisz and K. J. Christensen. Adaptive sampling methods to determine network traffic statistics including the hurst parameter. In *Proc. IEEE LCN '08*, Boston, USA, Oct 1998.
9. N. Duffield. Sampling for passive internet measurement: A review. *Statistical Science*, 19:472–498, 2004.
10. N. Duffield, C. Lund, and M. Thorup. Properties and prediction of flow statistics from sampled packet streams. In *Proc. of ACM SIGCOMM IMW '02*, Marseille, France, Nov 2002.
11. N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *SIGCOMM Comput. Commun. Rev.*, 30(4):271–282, 2000.
12. C. Estan and G. Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. Comput. Syst.*, 21(3):270–313, 2003.
13. A. Este, F. Gringoli, and L. Salgarelli. On the stability of the information carried by traffic flow features at the packet level. *SIGCOMM Comput. Commun. Rev.*, 39(3):13–18, 2009.
14. WAND Network Research Group. Auckland-vi traces. `http://www.wand.net.nz/wits/auck/6/auckland_vi.php`.
15. E. A. Hernandez, M. C. Chidester, and A. D. George. Adaptive sampling for network management. *J. Netw. Syst. Manage.*, 9(4):409–434, 2001.
16. H. Jiang, A. W. Moore, Z. Ge, S. Jin, and J. Wang. Lightweight application classification for network management. In *Proc. of ACM SIGCOMM INM '07*, Kyoto, Japan, Aug 2007.
17. A. Kumar and J. Xu. Sketch guided sampling - using on-line estimates of flow size for adaptive data collection. In *IEEE INFOCOM '06*, Barcelona, Spain, April 2006.
18. J. Mai, C. Chuah, A. Sridharan, T. Ye, and H. Zang. Is sampled data sufficient for anomaly detection? In *Proc. ACM SIGCOMM IMC '06*, Rio de Janeriro, Brazil, Oct 2006.
19. M. Mellia, M. Meo, L. Muscariello, and D. Rossi. Passive analysis of tcp anomalies. *Elsevier Computer Networks*, 52(14), October 2008.
20. A. Moore, D. Zuev, and M. Crogan. Discriminators for use in flow-based classification. Technical report, University of Cambridge, Computer Laboratory,, 2005.
21. T. Mori, M. Uchida, R. Kawahara, J. Pan, and S. Goto. Identifying elephant flows through periodically sampled packets. In *Proc. of ACM SIGCOMM IMC '04*, Taormina, Italy, 2004.
22. I. Paredes-Oliva, P. Barlet-Ros, and J. Solé-Pareta. Portscan detection with sampled netflow. In *Traffic Measurement and Analysis (TMA), Springer-Verlag LNCS 5537*, May 2009.
23. V. Paxson. End-to-end routing behavior in the internet. *SIGCOMM Comput. Commun. Rev.*, 26(4):25–38, 1996.
24. A. Pescapé. Entropy-based reduction of traffic data. *Communications Letters, IEEE*, 11(2):191–193, Feb. 2007.
25. B. Ribeiro, D. Towsley, T. Ye, and J. C. Bolot. Fisher information of sampled packets: an application to flow size estimation. In *Proc. of ACM SIGCOMM '06*, Rio de Janeriro, Brazil, 2006.
26. D. Rossi, C. Casetti, and M. Mellia. User patience and the web: a hands-on investigation. In *IEEE Globecom'03*, San Francisco, CA, USA, December 2003.
27. Zseby T. Deployment of sampling methods for sla. validation with non-intrusive measurements. In *Proc. of PAM '02*, Fort Collins, Colorado, USA, Mar 2002.
28. S. Valenti, D. Rossi, M. Meo, M.Mellia, and P. Bermolen. Accurate and fine-grained classification of p2p-tv applications by simply counting packets. In *Traffic Measurement and Analysis (TMA), Springer-Verlag LNCS 5537*, pages 84–92, May 2009.
29. T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall. Sampling and Filtering Techniques for IP Packet Selection. RFC 5475 (Proposed Standard), Mar 2009.