On the Impact of Sampling on Traffic Monitoring and Analysis[*]

Antonio Pescapé [ø], Dario Rossi [†],
Davide Tammaro [ø†], and Silvio Valenti [†]
[ø] Università di Napoli Federico II, Italy pescape@unina.it
[†] Telecom ParisTech, France, firstname.lastname@enst.fr

[*] This is the author version of the paper accepted to be presented to the 22nd International Teletraffic Congress (ITC-22) in Amsterdam, The Netherlands, on September 7-9, 2010.

***Abstract*—Due to significant advances in transmission technology and to the corresponding increase of link rates, traffic sampling is becoming a normal way of operation in traffic monitoring. Given this trend, in this paper we aim to assess the impact of the sampling 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 different sampling rates. To quantify the amount of degradation and bias that sampling introduces with respect to the unsampled traffic we use well-known statistical measures (i.e., Hellinger Distance, Fleiss Chi-Square). Unlike previous work, we consider a very large set of "features" (i.e., any kind of properties characterizing 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 typically exploited by a rather wide class of applications, such as traffic monitoring, analysis, accounting, and classification. Using three real traffic traces, representative of different operational networks, we find that (i) a significant degradation affects a wide number of features; (ii) the set of features less degraded is consistent across the three datasets; (iii) at the same time, some artifacts may arise, resulting in lower distortion scores at higher sampling rates, which are tied to both the specific metric, as well as the way in which the feature is computed (e.g., binning); (iv) no significant reduction of the estimation bias can be obtained by merely tweaking the sampling policy – which partly contrasts earlier observations concerning the better quality achievable with stratified sampling.**

I. 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. Although *data aggregation* is a core technique in current Internet, as the widespread use of Simple Network Management Protocol (SNMP) 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: a number of studies focus on the design of sampling policies and on their impact, typically considering a few metrics only [1]–[9], whereas other works concentrate on a single application of sampling [10]–[15].

In this paper we aim at quantifying the robustness of a large number of properties characterizing the traffic flows (i.e. "features") under different sampling strategies. 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 features is evaluated in terms of statistical indexes (i.e. "metrics"), such as the *Hellinger Distance* and the *Fleiss Chi-Square* of sampled versus unsampled data. The proposed methodology is based on a popular flow-level analyzer, Tstat [16], which operates on packet level traffic producing a wealth of statistical features, and which we instrumented with 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 as to assess the level of degradation introduced by sampling.

We perform experiments on three real traffic traces, some of which are publicly available, so that our investigation is representative of rather heterogeneous scenarios. Our results show that a substantial degradation affects the majority of features already for low levels of sampling. Yet our analysis highlights that the distortion may vary for features pertaining to different protocol layers: indeed, properties whose estimation relies on the inspection of a single packet (e.g., IP or UDP properties) are generally less distorted than properties depending on multiple packets (e.g., inter-arrival, RTT, etc.). At the same time, we also find the TCP case to be more complex, as single-segment properties often require *specific* segments to be sampled (e.g., those negotiating specific options), and are as a result severely affected already at low sampling rates.

Moreover, we are able to individuate a set of features "robust" to sampling (i.e., minimally distorted), which is furthermore consistent across all the datasets. By focusing on such a reduced subset of features, we perform a thorough sensitivity analysis and find that no sampling policy is able to reduce the distortion induced by sampling. This is an interesting finding, that partially counters earlier observations (focusing on a narrower set of features, i.e., mainly traffic volume) concerning the better quality achievable with stratified sampling, and which we believe to be tied to the level of statistical multiplexing already present in the traces.

Incidentally, we also point out some unexpected behavior of some features, whose distortion apparently decreases when the sampling rate increases. Digging further, we find the root cause of this phenomenon to be the joint effect of the type of traffic, the distortion metric used and the features estimation procedure. This suggest that extra care must be taken when dealing with sampled traffic, as otherwise uncorrelated factors may combine together and yield misleading conclusions.

The reminder of this paper is organized as follows. In Sec. II we overview the most relevant work, highlighting the relations with our study and describing the main contributions of this work. We describe the followed methodology in Sec. III, 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. IV. Finally, conclusive remarks and future directions are discussed in Sec. V.

II. 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 [17], 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 [18]. 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 [18]. 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 [4], proposing then more sophisticated techniques based on hash functions [19], sample and hold [20], and hash-based sketches [7]. Other works focused instead on making the sampling rate *adaptive* [3], [10], [21], for instance to the traffic load.

Major results can be summarized from the above works. First, authors of [4] 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 [17]) has been also pointed out in [4], [22], although more recent results [9] suggest that the statistical multiplexing of traffic can have the same effect of a random selection process. In fact, [9] 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 [19], sketches for [7] flow-size and so on. Moreover, most work to date focuses on specific metrics, essentially accounting for traffic volumes under sampling [5]–[9]. More recently researchers have started investigating the impact that sampling may have on a wider range of applications, such as network management [10], SLA verification [11], traffic classification [12], [23] or anomaly detection [13]–[15]. 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 [5]–[9] 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 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, [24] 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 [24] focuses again on a specific application, namely traffic classification). In [23] another closer contribution to ours is proposed: mainly, obtained results indicate that the accuracy of standard classification 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.

To highlight the significance of the contributions of this work we underline that, to the best of our knowledge, it extends the results present in literature in that: (i) it is one of the first attempt to study the impact of sampling on a very broad set of traffic features (see Tab. I); (ii) we found a very limited number of features can be safely estimated under sampling; (iii) we found the way packet sampling is performed has a very limited impact on the estimation accuracy when a large set of features is considered.

III. Methodology

First, we elaborate on the *features* (Sec. III-B) we focus on. We then describe the *sampling policies* (Sec. III-B) we take into account, as well as the different statistical *metrics* (Sec. III-C) used to evaluate the distortion induced by sampling. Finally, we briefly describe the *datasets* (Sec. III-D) used throughout this work.

A. Features

Tstat [16] 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 of the measurement (e.g., incoming versus outgoing versus local, and client-2-server versus server-2-client). A summary of such properties is reported in Tab. I, divided according to (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 [27], which contains the most relevant features for traffic classification. Yet, we point out that our work uses these features with a different semantic from [27], as we consider the feature distortion mostly in its *aggregated* form, whereas traffic classification needs measures at an *individual* flow level – an interesting aspect we leave for future work.

B. Sampling Policies

We implement different sampling policies as defined [18]. For the time being, we have implemented "unbiased" sampling techniques, leaving biased techniques as a future work. 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**: k consecutive packets are grouped in a window, in which a single packet is randomly sampled.

C. 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

TABLE I

LIST OF CONSIDERED FEATURES. STAR SIGN (⋆) DENOTES FEATURES MEASURED FOR INCOMING VS. OUTGOING VS. LOCAL DIRECTIONS.

IP (single datagram)	ip_tos⋆	TOS field
	ip_ttl⋆	TTL field
	ip_len⋆	Packet length [byte]
	ip_bitrate⋆	Bitrate [kbit/s]
	ip_protocol⋆	Protocol type
UDP (single segment)	udp_port_flow_dst	Destination port per flow
	udp_port_dst⋆	Destination port per segment
	udp_tot_time	Flow lifetime [ms]
	udp_cl_b_l⋆	Flow length [byte], coarse granularity
	udp_cl_b_s⋆	Flow length [byte], fine granularity
	udp_cl_p⋆	Flow length [packet]
TCP (single segment)	tcp_mss_used	Negotiated MSS
	tcp_mss_b	MSS declared by Server
	tcp_mss_a	MSS declared by Client
	tcp_opts_TS	Timestamp option
	tcp_opts_WS	WindowScale option
	tcp_opts_SACK	SACK option
	tcp_bitrate⋆	Application bitrate
	tcp_port_syndst⋆	Destination port (SYN segments only)
	tcp_port_synsrc⋆	Source port (SYN segments only)
	tcp_port_dst⋆	Destination port (all segments)
	tcp_port_src⋆	Source port (all segments)
TCP (mutliple segments)	tcp_interrupted	Early interrupted flows [25]
	tcp_thru ⋆	Application throughput [Kbps]
	tcp_tot_time	Flow lifetime
	tcp_rtt_cnt	RTT: number of samples
	tcp_rtt_stdev	RTT: standard deviation [ms]
	tcp_rtt_max	RTT: maximum RTT [ms]
	tcp_rtt_avg	RTT: average RTT [ms]
	tcp_rtt_min	RTT: minimum RTT [ms]
	tcp_cl_b_l	Flow length, coarse granularity [byte]
	tcp_cl_b_s	Flow length, fine granularity [byte]
	tcp_cl_p	Flow length [packet]
	tcp_cwnd	TCP in-flight-size [byte]
	tcp_win_max	TCP max RWND [byte]
	tcp_win_avg	TCP average RWND [byte]
	tcp_win_ini	TCP initial RWND [byte]
	tcp_anomalies ⋆	TCP anomalies as defined in [26]
RTCP (mutliple segments)	rtcp_bt⋆	Average bitrate [bit/s]
	rtcp_mm_bt⋆	Associated MM flow bitrate[kbit/s]
	rtcp_mm_cl_b⋆	Associated MM flow length [bytes]
	rtcp_mm_cl_p⋆	Associated MM flow length [packets]
	rtcp_t_lost⋆	Lost packets per flow
	rtcp_f_lost⋆	Prob. of lost packets
	rtcp_dup⋆	Duplicated packets
	rtcp_lost⋆	Lost packets
	rtcp_avg_inter⋆	Average inter-packet gap (IPG)
	rtcp_jitter⋆	Average jitter
	rtcp_rtt⋆	RTCP Round trip time [ms]
	rtcp_cl_b⋆	RTCP flow length [bytes]
	rtcp_cl_p⋆	RTCP flow length [packets]
RTP multimedia (mutliple segments)	mm_burst_loss⋆	Burst length of lost packets [packet]
	mm_p_late⋆	Prob. of late packets
	mm_p_lost⋆	Prob. of lost packets
	mm_p_dup⋆	Prob. of duplicate packets
	mm_p_oos⋆	Prob. of out-of-sequence packets
	mm_n_oos⋆	Length of out-of-sequence burst
	mm_oos_p⋆	Total out-of-sequence packets
	mm_reord_p_n⋆	Total reordered packets
	mm_reord_delay⋆	Delay of reordered packets
	mm_avg_jitter⋆	Average jitter [ms]
	mm_avg_ipg⋆	Average IPG [ms]
	mm_avg_bitrate⋆	Stream bitrate [kbit/s]
	mm_cl_b⋆	Long stream flow length [bytes]
	mm_cl_p⋆	Long stream flow length [packet]
	mm_cl_b_s⋆	Short stream flow length [bytes]
	mm_cl_p_s⋆	Short stream flow length [packet]
	mm_tot_time_s⋆	Short stream flow lifetime [ms]
	mm_tot_time⋆	Stream flow lifetime [s]
	mm_rtp_pt⋆	RTP payload type
	mm_uni_multi⋆	Unicast/multicast flows
	mm_type⋆	Stream type

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:

- **Fleiss Chi-Square (ϕ)**

$$\phi(p, q) = \sqrt{\frac{\sum_{x \in X} [q(x) - p(x)]^2 / p(x)}{\sum_{x \in X} [q(x) + p(x)]}} \qquad (1)$$

- **Hellinger Distance (HD)**

$$HD(p, q) = \sqrt{1 - \sum_{x \in X} \sqrt{p(x) q(x)}} \qquad (2)$$

To provide backward compatibility with [4], we consider the ϕ metric, which is a normalized version of the standard Chi-Square widely used also, e.g., for classification purposes [28]. As the Chi-Square statistic is sensitive to the size of the data set, this makes it difficult to compare samples of varying sizes: thus, it cannot quantify significant trends when varying the sampling fraction. Fleiss' definition of ϕ directly derives from Chi-Square but overcomes this limitation, being independent from the sample size [4].

The Hellinger Distance (HD) is typically used as a score of similarity between metrics, and it has been used in [29] to assist the context of classification as well. HD values are confined in the range $[0, 1]$, with lower values corresponding to higher similarity between the distribution under comparison. An extended set of results is available in [30], which also consider other metrics, such as Kullback-Leibler, used e.g., in [31] to reduce the data set size in an approach complementary to sampling.

D. 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. II. Namely, the top portion of 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, representative 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 (Voice, Video/TV, Data) over broadband access. ISP is representative of a very heterogeneous scenario, in which no traffic restriction are applied to customers.
- **Auckland-VI** is continuous 4.5-days long trace captured during 2001 at the Internet egress router of the University of Auckland, publicly available at [32].

To preliminarily assess the amount of traffic to which our investigation refers to, we investigate the *Flow-Recall (FR)* and *Byte-Recall (BR)* induced by sampling. Specifically, we define *FR* as the percentage of flows whose packets are selected by sampling, and *BR* as the correspondent percentage of bytes carried by flows which are selected by sampling (note that this metric takes into account all packets of those flows of

TABLE II
SUMMARY OF DATASET USED IN THIS WORK.

Trace	ISP	Campus	Auckland-VI
Year	2006	2008	2001
Packets	44,396,297	17,246,459	291,052,998
Flows	219,481	422,928	11,128,910
Packets/flow	202.27	40.77	26.15
IPs	61,959	81,687	410,059
FR ($k = 2$)	1.125	0.938	1.130
BR ($k = 2$)	0.992	0.934	0.999
FR ($k = 128$)	0.197	0.138	0.136
BR ($k = 128$)	0.943	0.727	0.687

which at least one packet has been sampled). As an example, bottom portion of Tab. II reports *FR* and *BR* results considering two different sampling rates ($k = \{2, 128\}$) for the uniform sampling policy for all dataset. As it can be seen, at low sampling step $k = 2$, the number of flows artificially inflates for the ISP and Auckland traces: as already observed in [5], long flows can be split if the time between sampled packets exceeds the flow timeout (which defaults 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 recall is always very high, meaning that results reported in this paper are representative of the bulk of traffic. Clearly, the *BR* metric is tied to the average number of packets constituting a flow (reported in top portion of Tab. II), as the longer the flows, the higher the byte recall.

IV. EXPERIMENTAL RESULTS

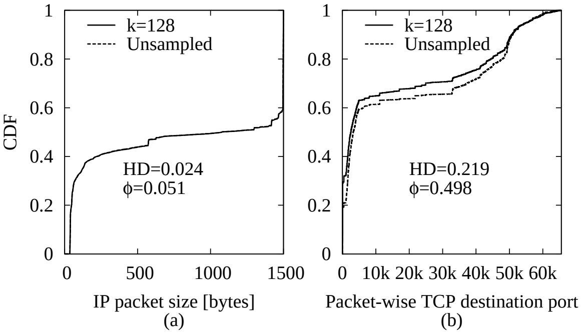

Fig. 1. Aggregate level of Campus trace: CDF of IP packet size (a) and number of packets per destination TCP port (b). Plots report the CDF gathered from the unsampled vs sampled traffic aggregate, along with the statistical indexes of distortion.

In this section, we first analyze the *range of the variation* of the selected metrics and features (Sec. IV-A). Then, we analyze the behavior of *all features* grouped by protocol layer under increasing sampling rates but focusing mainly on uniform sampling (Sec. IV-B). This analysis allows us to select a set of *robust features* (i.e. less distorted across all datasets), on which we conduct a thorough *sensitivity analysis* by applying a wider range of sampling policies and rates (Sec. IV-C).

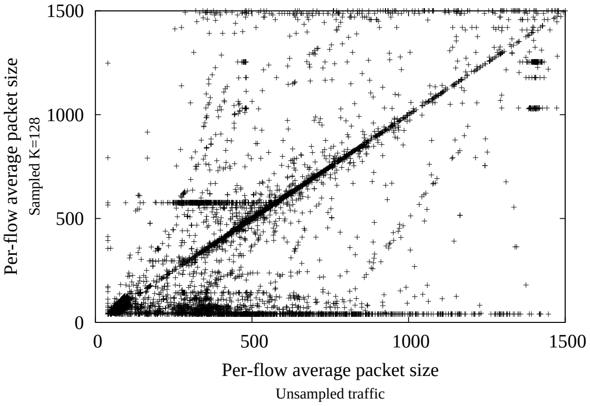

Fig. 2. Individual-flow level: scatter plot of the unsampled versus sampled average per-flow IP packet size.

A. Playing with Distortion Scores

To have a first idea of the scale of the distortion scores defined so far we provide a preliminary example of some relevant features. With reference to Campus trace Fig. 1-(a) and Fig. 1-(b) report the CDF of two features, respectively counting the IP packet size in bytes and the number of packets directed to a given TCP port. 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 shows a moderate distortion, with a corresponding degradation of $HD = 0.219$ and $\phi = 0.498$: in this case, differences in the CDF, although modest, can be seen with naked-eyes from the plot. Conversely, IP packet size shows a degradation score of about one order of magnitude smaller for both metrics $HD = 0.024$ and $\phi = 0.051$: 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 features between the two shown in Fig. 1, 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 example gives us some preliminary insights on the relationship between the two observation levels. Fig. 2 shows a scatter plot of the sampled versus unsampled metrics whose CDF is shown in Fig. 1-(a). 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 by observing a single sample, it is likely to get a typical-sized packet (e.g, a $40-$byte packet without

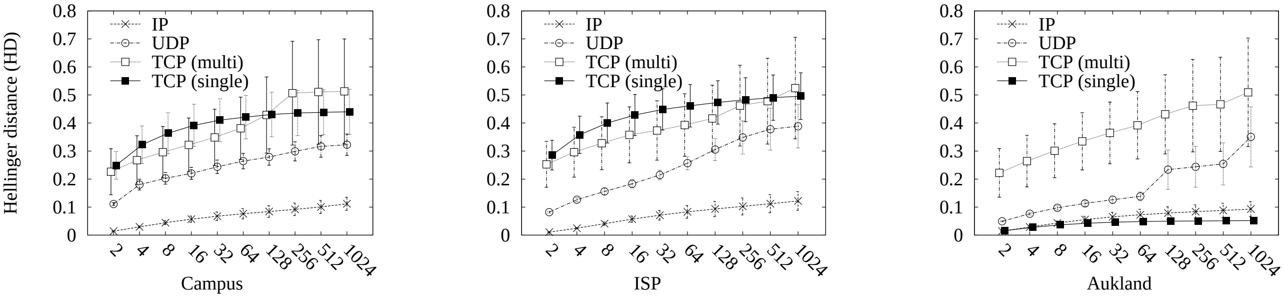

Fig. 3. Mean and variance of the HD distortion score, for features grouped by protocol layer, as a function of the sampling step under uniform sampling policy.

data, or $1500-$byte full payload packet, or a $576-$byte packet) which will lead to a bad estimation of the actual average packet size of the flow (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.

B. Protocol Layer Impact

To refine our understanding of sampling impact on a large number of features, we start by grouping the features in different sets according to the protocol layer: in particular we consider IP features, UDP single-segment features, TCP single- and multiple- segment features as in Tab. I. By comparing the effect of sampling on these groups, we want to find out whether there exists a family of features which is by definition more robust to sampling.

Without loss of generality, for the time being we express the distortion scores using the Hellinger Distance. We also select a single sampling policy (namely, uniform sampling) and consider sampling rates ranging from 1/2 to 1/1024. Results are reported in the three graphs of Fig. 3, which correspond to the different datasets. In every single plot, each curve depicts the mean and the variance of the HD metric over a given group of features as a function of the sampling step k.

As a first general comment, it can be seen that the distortion score for the different groups exhibits, with minor exceptions, a consistent behavior across datasets. In other words, there are features that are intrinsically easier to quantify under sampling: for instance, features relying only on the inspection of a single packet (e.g., IP packet size) can be expected to be more robust to sampling than features depending on the observation of multiple packets (e.g., inter-arrival time). This intuition is confirmed by the plots, where the curves of distortion scores for both IP and UDP single-segment features are considerably closer to the minimum value for the HD.

The behavior of TCP features is instead more complex and counter-intuitive. In fact, notice that the trend of the two TCP groups of features varies for different datasets. Considering for example the Auckland trace, we observe that single-segment TCP features, which are directly derived from TCP header fields or options (e.g. related to MSS negotiation, window scale, etc.), exhibit an unusually low distortion score. Investigating this issue further, we found out that in the Auckland dataset, portions of the captured traffic are obfuscated (i.e., more precisely, set to zero) for privacy reasons. Incidentally, also the portion of the packet header carrying TCP options undergoes this obfuscation process, making Tstat unable to correctly measure the related features (i.e., more precisely, Tstat assumes a maximum value for MSS, and by default considers timestamp, window scale and sack options as unused). Therefore, in this case the low distortion score is an artifact, arising from impossibility of correctly estimating the features from the trace under investigation, even in the unsampled case.

On the other hand, notice that, apart from of the Auckland dataset, at lower sampling rates, TCP features depending on multi-segment suffer a smaller distortion than that of TCP features depending on single-segment observation. Also, the HD value for TCP multi-segment features keeps increasing with the sampling, whereas TCP single-segment features, albeit already distorted for low levels of sampling, do not further degrade for high sampling factors. This unexpected behavior is due to the fact that, in the TCP case, some of the single-segment features require *specific segments* to be monitored: for instance the segment corresponding to the negotiation of a specific option. If this very segment is missed because of sampling, which is often the case already at low sampling rates, the features estimation is compromised. Conversely, some of the features requiring multiple segments (e.g., average and maximum value of the receiver window, etc.) can still be safely estimated for low sampling rates.

C. Sensitivity analysis

In this section we investigate the impact of different sampling policies and rates as well as the use of different metrics to express the distortion. Under this wider perspective, we aim at isolating a set of features which are more robust to sampling (or, equivalently, less distorted), irrespectively of the protocol layer they pertain to. Therefore, we first define a “robustness” criterion to identify such features. Then, by focusing on this reduced set, we perform a more detailed sensitivity analysis.

1) Robust feature set: To identify the robust set of features we employ a simple threshold-based criterion based on the Hellinger Distance: features whose HD value is lower than the defined threshold are considered robust. More specifically in the following results refers to features which have an $HD < 0.1$ with a sampling of $k = 128$, but similar considerations hold for other values of the threshold (cfr. [30]) as well. Notice

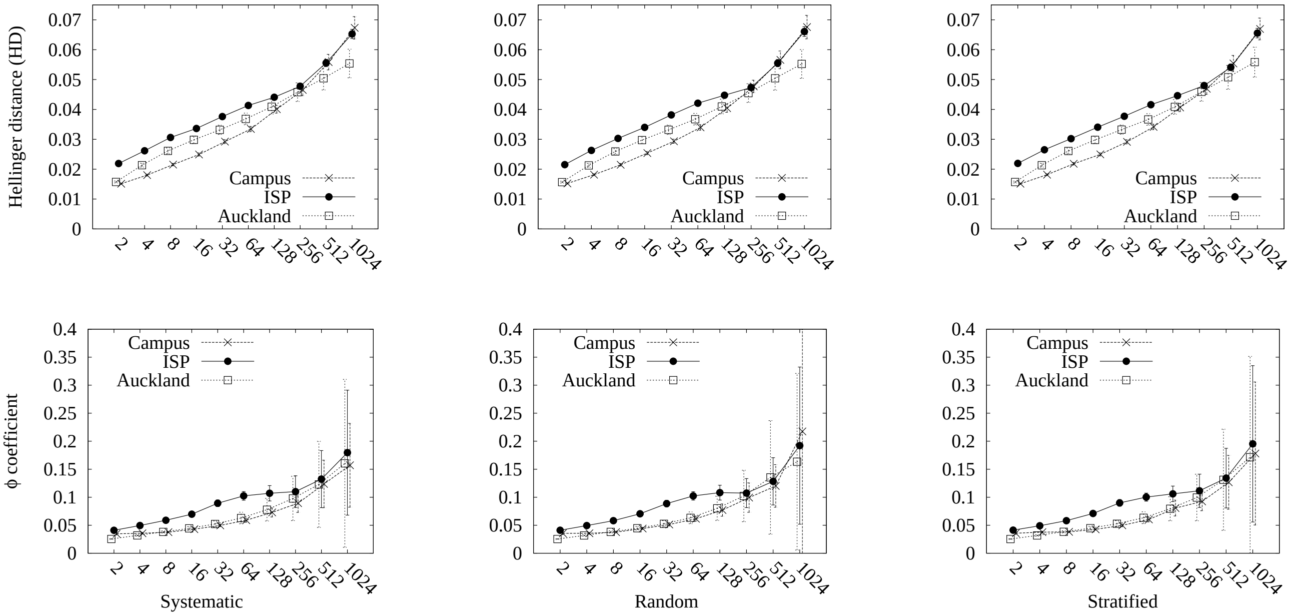

Fig. 4. Sensitivity analysis: Mean and variance of the statistical distortion scores of the robust features set, for different datasets, sampling policies and rates.

that we select $HD = 0.1$ in reason of the existence of a clear separation between the curves in Fig. 3. Moreover notice also 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.

It is important to stress that we no longer take into account the grouping by protocol layer when applying the robustness criterion. Rather, features are evaluated individually, so that the robust set actually consists of properties belonging to different groups. As we also consider each direction separately (i.e., incoming versus outgoing versus local traffic), it may happen that a feature is robust for a given direction, but not for the opposite one. Moreover, we conservatively require features to be *jointly* robust across all datasets under consideration: in other words, the resulting set is the *intersection* of the sets of robust features on each single datasets.

The final set contains 34 features, 10 of which belong to the IP layer (representing the 66.6% of the 15 IP layer features computed by Tstat), 20 of which belong to the TCP layer (16.7% of the TCP features) and the remaining 4 to the UDP layer (23.5%). Thus, each protocol layer is represented in the robust set, except for the RTCP and MM layers which are missing. In fact, the relatively low amount of MM/RTCP traffic present in the Auckland dataset makes it difficult to evaluate the related features for this traces, especially when hard sampling conditions further limit the number of valid samples.

As for the *union* of the robust feature sets of each single trace, such set contains 110 features, and it is larger than the intersection. This means that the actual amount of distortion experienced by features may also depend on the dataset – suggesting that some features are robust only under specific traffic conditions.

2) Impact of Sampling Policy: In this section we investigate on the robust set of features just defined: for lack of space and to avoid cluttering the overall picture, we omit the analysis of the complete feature set, which is available in [30].

Results of the sensitivity analysis for the robust set are reported in Fig. 4: graphs are arranged in a matrix, whose columns correspond to the different sampling policies, while rows are related to the two statistical metrics used to quantify the feature distortion. As before, for each sampling policy, we employ an exponentially increasing sampling step $k = 2^i, i \in [1 \ldots 10] \subset \mathbb{N}$, reported on the x-axis of every plot. Each graph contains three curves, one for each dataset, depicting the average distance score over the 34 features belonging to the robust set; variance of the distance score is also reported, by means of vertical error bars (notice that stdev is visually noisy since the square root of score values in $[0, 1] \in \mathbb{R}$ exploses).

At first glance, we can observe that there is no clear advantage over the choice of stratified sampling over random or systematic sampling: indeed considering either row of three plots, one can gather a striking similar behavior. This finding holds whenever several features are considered, and contrasts with earlier results supporting stratified sampling techniques [4]. Our intuition is that, given the level of statistical multiplexing of traffic flows, the sampling policy has a minor impact, especially when complex traffic properties are considered. Also, notice that similar conclusions have been recently reported by independent research [9], which however limitedly considers only traffic volume measurements under sampling (i.e., flow length).

Let us now compare the different distortion scores and focus on the two graphs belonging to the same column, so considering each single sampling policy on its own. Although the two metrics take values in different ranges, they both identify the Campus and Auckland dataset as the best case yielding lower distortion scores, and agree that ISP constitutes a stiffer scenario under sampling. However another important consideration stems from the comparison of the plots of the

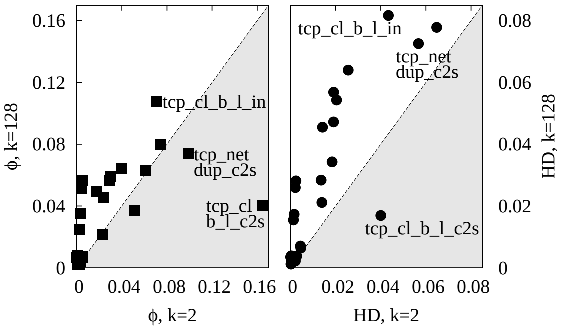

Fig. 5. Scatter plot of HD and ϕ distortion for the robust feature subset, considering sampling $k = 2$ and $k = 128$.

two distortion metrics. For HD first, one can notice that the range of distortion values is similar for all datasets, and that a monotonously increasing slope characterizes all curves. Instead, as far as the ϕ score is concerned, such behavior is observed only for the Campus and Auckland traces, but not for ISP one. Indeed, in this case, the curve shows a almost flat portion for sampling rates in the range $k = [64, 256]$: this is an interesting point that deserves further attention, and that we investigate further in the following.

3) Artifacts of Distortion Score: To better understand the phenomenon early observed in Fig. 4, we need to focus more closely on the features that are robust across all datasets. For this purpose we resort to the scatter plots of Fig. 5. Each robust feature is represented by a single point whose (x,y) coordinates are the distortion scores for $k = 2$ and $k = 128$, where the flatness of the ϕ score is observed. The left plot reports the value of the ϕ coefficient while the right one is related to the HD distance.

In the picture, we label some representative points 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. Instead, some features exhibit an opposite and counter-intuitive behavior, falling in the gray-shaded area which correspond to features whose distortion score actually *reduces* with higher sampling rates. For instance, this effect is particularly evident for the `tcp_cl_b_l_c2s` feature, i.e. the TCP flow length, 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 the improvement of the feature estimation is a joint consequence of the traffic nature (sampling tends to select packets from the same elephant flows, yielding a better estimation of the length of such flows) and the specific binning adopted (as this behavior is not shown by the corresponding feature calculated with fine granularity `tcp_cl_b_s_c2s`).

Notice that this effect is instead less evident in the HD score plot, where only a single feature falls in the gray region, than in the ϕ plot where we actually find three points in this area. Moreover for the ϕ coefficient many features actually fall on the bisector as well, which means that no degradation is detected by the distance metric despite the increased sampling rate. In fact, it seems as though different choices of binning have a greater impact on the ϕ metric, sometimes compromising its accuracy. On the other hand, the HD distance appears able to better characterize the distortion, because a greater score usually corresponds to a larger sampling step. This is due to the different weighting of the errors in ϕ and HD: in the former, larger discrepancies will be amplified (i.e., squared difference) with respect to the latter score (i.e., product): this entails that several small errors, affecting several bins, may produce a lower distortion score in ϕ. The main outcome of this reflexion is that special care must be also taken in the selection of the distortion metric used, as otherwise similar artifacts may yield to misleading conclusions.

V. Discussion and 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, (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 two statistical indexes, apt at quantifying the amount of features degradation introduced by sampling.

Our results show that, on the one hand, sampling causes an important degradation of the features estimation: indeed, most of the features are already severely distorted at low sampling rates. By separately analyzing properties belonging to different protocol layers, we find that a lower level of distortion affects the features based on the estimation of a single packet (e.g., those related to IP or UDP) with respect to those related to more packets, except in the case of features whose estimation rely on very specific segments (e.g., as for some TCP features).

On the other hand, we have found that, irrespectively of the protocol layer considerations, there exists a small set of features 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 only has a minor impact on reducing the degradation. 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); second, this evidence may have been hidden by previous work which typically focused on a few specific features only (e.g., traffic volumes). At the same time, we have also isolated a number of counter-intuitive behaviors and measurement artifacts, showing that it may be challenging to correctly assess the impact of sampling even on simple measures.

In future work, we aim at extending this study 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 may improve the estimation of some features (e.g., SYN sampling is useful for flow-length, while batch-sampling is useful for packet inter-arrival, etc.), and as such their impact on other features is worth investigating as well: eventually, results may suggest that several sampling process, each optimized to monitor a specific feature, shall be run in parallel to gather consistent results over all features of interest. Second, we aim at considering a wider range of applications (e.g., traffic classification, anomaly detection, etc.) so as to better correlate the feature distortion with the performance of the application itself. In particular, as early work [12], [15] 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 improvement 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.

VI. Acknowledgments

This work was funded by the Celtic project TRANS and under COST TMA Action IC070.

References

[1] Luigi Alfredo Grieco and Chadi Barakat. An analysis of packet sampling in the frequency domain. In *IMC '09: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 170–176, New York, NY, USA, 2009. ACM.

[2] P.D. Amer and L.N. Cassel. Management of sampled real-time network measurements. In *Proc. of IEEE LCN '89*, Oct 1989.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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 Janeiro, Brazil, 2006.

[9] 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.

[10] E. A. Hernandez, M. C. Chidester, and A. D. George. Adaptive sampling for network management. *J. Netw. Syst. Manage.*, 9(4):409–434, 2001.

[11] Zseby T. Deployment of sampling methods for sla. validation with non-intrusive measurements. In *Proc. of PAM '02*, Fort Collins, Colorado, USA, Mar 2002.

[12] 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.

[13] 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 Janeiro, Brazil, Oct 2006.

[14] 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 Janeiro, Brazil, Oct 2006.

[15] 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.

[16] Tstat, http://tstat.tlc.polito.it.

[17] N. Duffield. Sampling for passive internet measurement: A review. *Statistical Science*, 19:472–498, 2004.

[18] 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.

[19] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *SIGCOMM Comput. Commun. Rev.*, 30(4):271–282, 2000.

[20] 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.

[21] 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.

[22] V. Paxson. End-to-end routing behavior in the internet. *SIGCOMM Comput. Commun. Rev.*, 26(4):25–38, 1996.

[23] 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.

[24] 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.

[25] 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.

[26] M. Mellia, M. Meo, L. Muscariello, and D. Rossi. Passive analysis of tcp anomalies. *Elsevier Computer Networks*, 52(14), October 2008.

[27] A. Moore, D. Zuev, and M. Crogan. Discriminators for use in flow-based classification. Technical report, University of Cambridge, Computer Laboratory,, 2005.

[28] A. Finamore, M. Mellia, M. Meo, and D. Rossi. Kiss: Stochastic packet inspection. In *Traffic Measurement and Analysis (TMA), Springer-Verlag LNCS 5537*, pages 117–125, May 2009.

[29] 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.

[30] Antonio Pescape, Dario Rossi, Davide Tammaro, and Silvio Valenti. On the impact of sampling on traffic monitoring and analysis. http://www.enst.fr/ drossi/paper/rossi10techrep.pdf, 2010.

[31] A. Pescapé. Entropy-based reduction of traffic data. *Communications Letters, IEEE*, 11(2):191–193, Feb. 2007.

[32] WAND Network Research Group. Auckland-vi traces. http://www.wand.net.nz/wits/auck/6/auckland_vi.php.