INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 0000; **00**:1–26
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nem

Exploiting packet sampling measurements for traffic characterization and classification

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

[1]*INFRES Department, TELECOM ParisTech, 46 rue Barrault, 75634 Paris, France*
[2]*Department of Computer Science and Systems, University of Naples Federico II, Via Claudio 21, 80125 Napoli, Italy*

SUMMARY

The use of packet sampling for traffic measurement has become mandatory for network operators to cope with the huge amount of data transmitted in nowadays networks, powered by increasingly faster transmission technologies. Therefore, many networking tasks must already deal with such reduced data, more available but less rich in information. In this work we assess the impact of packet sampling on various network monitoring activities, with a particular focus on traffic characterization and classification. We process an extremely heterogeneous dataset composed of four packet level traces (representative of different access technologies and operational environments) with a traffic monitor able to apply different sampling policies and rates to the traffic and extract several features both in aggregated and per-flow fashion, providing empirical evidences of the impact of packet sampling on both traffic measurement and traffic classification. First, we analyze feature distortion, quantified by means of two statistical metrics: most features appear already deteriorated under low sampling step, no matter the sampling policy, while only a few remain consistent under harsh sampling conditions, which may even cause some artifacts undermining the correctness of measurements. Second, we evaluate the performance of traffic classification under sampling. The information content of features, even though deteriorated, still allows a good classification accuracy, provided that the classifier is trained with data obtained at the same sampling rate of the target data. The accuracy is also due to a thoughtful choice of a smart sampling policy which biases the sampling towards packets carrying the most useful information.

Received . . .

KEY WORDS: packet sampling; traffic classification

1. INTRODUCTION

Even if the success of the Internet as a global communication platform is greatly due to its decentralized and open architecture, the advances in communication and commutation technology have played an equally important role, supporting new generations of bandwidth-intensive applications. As a side effect of this increasing transmission speed, network operators must deal with ever growing traffic and the consequent huge amount of measurements, extremely challenging to collect, store and process. Therefore, *packet sampling* has become mandatory for effective passive network measurements, especially in the core of the network, to reduce the amount of data to a manageable size. Naturally such reduction comes at the cost of less accurate data and several studies have focused on the impact of different sampling policies on traffic measurement [1–8] or on the performance of various networking activities related to measurement, such as monitoring, SLA compliance, anomaly detection and traffic classification [9–15].

*Correspondence to: INFRES Department, TELECOM ParisTech, 46 rue Barrault, 75634 Paris, France, E-mail: silvio.valenti@telecom-paritech.fr

Among the various applications of traffic measurement, traffic classification has recently received significant attention from the research community. Correctly identifying the application associated to traffic flows is indeed crucial, in that this knowledge is necessary for a number of management tasks (e.g., differential treatment for quality of service, or service level agreements enforcing to cite only the most important). Inevitably, as sampled data is becoming in many settings the only kind of data available, the question is whether classification is still possible with such reduced information. Still, despite this growing interest in the operation community, the impact of packet sampling and the performance of traffic classification techniques are issues that have been rarely considered together in the literature so far [14–20].

Given these premises, this paper provides two main contributions. First, we measure the effect of different sampling policies and rates on a large set of traffic properties, without being tied to any particular application. Second, we focus on traffic classification under sampled data (hence, traffic properties becomes "features", in machine learning terms, upon which classification decisions are taken): in this context, we evaluate the extent of the degradation of the information content conveyed by the features, besides investigating the achievable classification accuracy.

In order to gather general results, we employed an extremely heterogeneous dataset, composed by four different traces, from both academic and commercial networks, representative of different access technologies, network setting, user population and years. When possible we used publicly available dataset to promote cross-comparison in the scientific community. This dataset was processed by a popular flow-level analyzer, `tstat` [21], which outputs not only detailed per-flow features at network and transport layer, but also aggregated distributions of feature. For this work, we enhanced the tool with the possibility of applying several sampling policies (namely, *systematic*, *uniform*, *stratified* and *biased*) with arbitrary sampling rates.

We characterize the distortion introduced by sampling as the distance between the distribution of properties for sampled and unsampled traffic, by means of two commonly used statistical metrics, namely the *Hellinger Distance* and the *Fleiss Chi-Square*. To summarize our main results, we find that (i) most properties are already distorted for low sampling rates, regardless of the sampling policy applied (partially confuting earlier results about the robustness of random sampling, which, however, limitedly considered only a small subset of properties) and on the other hand, we found a set of properties, pertaining to different layers, which are correctly estimated in spite of sampling. Also, (ii) we highlight some artifacts induced by heavy sampling, which seem to improve the estimation of some properties, that are due to a subtle interaction of binning strategies and the statistical metric used to compute the distortion score; besides, we observe that biased sampling yields a greater distortion, which makes it less suited for the purpose of traffic characterization.

As for the impact of the sampling on traffic classification, we rather use a biased sampling policy which tries to capture the most important pieces of information. For this task, we first asses the usefulness of features using a information theoretic metric, namely the *Information Gain*, and afterwards we evaluate the classification accuracy through C4.5, a widely know supervised classification algorithm. We study the impact of different sampling rates, feature groups, datasets and training policies. Our main findings are that (i) features, albeit distorted, are still quite useful for the classification (i.e., their information content is less altered); therefore, (ii) provided that both the training and validation phase are performed with data obtained applying the same sampling rate, it is still possible to have a good classification accuracy with sampled data.

The remainder of this work is organized as follows. First an overview of the related work is found in Sec. 2. Then, Sec. 3 presents the data used for the experimental part: first, the packet level traces and, second, the features produced by our monitoring tool. We then describe the methodology used to process this data in Sec. 4: in particular the sampling policies, the statistical metrics and the classification algorithm. Results are split in three sections: as for the impact of sampling on traffic measurement, Sec. 5 deals with the distortion of the *aggregate features* and Sec. 6 with the distortion of the *per-flow features*; as for traffic classification, Sec. 7 presents the impact of sampling on *traffic classification*. We summarize and conclude our paper in Sec. 8.

2. RELATED WORK

Packet sampling is not a novel technique [22]. Yet, given its increasing importance, it has recently received a lot of attention from the research community. In the following we overview the most important pieces of work on this topic, both on sampling itself and on its effect on traffic classification, in order to better highlight our contribution. Yet, this is far from being a complete survey, for which we rather refer the reader to [22].

First of all, researchers have categorized packet sampling methods in a few classes, starting from [1], until they have finally converged to a common framework standardized as an IETF RFC [23]. Summarizing, a first distinction can be made according to the selection scheme, which can be *deterministic*, *random* or *content-based*. Second, we can differentiate sampling techniques according to the selection trigger, based on the amount of *time* or number of *packets* between two different sampling events. As far as the selection scheme is concerned, researchers have demonstrated that the statistic properties of random sampling, especially in its stratified declination, make this technique particularly robust to evasion and attacks [3, 22, 24]. On the other hand, recent studies showed that statistical multiplexing of traffic may have the same effect of a random selection [8], especially when considering estimation of traffic volumes. There is much more of an agreement, instead, about the most effective selection trigger: [3] showed that time-based triggers are less robust than packets-based ones, because they suffer from the bursty nature of network traffic. A few works have proposed more sophisticated sampling techniques which help in estimating specific traffic features, for instance trajectory sampling for spatial properties [25] or sketches for flow-size [6]. Other works have proposed to make the sampling rate *adaptive* [2,9,26] for instance to the traffic load, to reduce the estimation error of some traffic metrics.

Besides investigating the properties of sampling itself and its impact on mostly traffic volumes measurements [4–8], researchers have also studied the possible applications of sampled data for various network administration tasks, such as network management [9], SLA verification [10], anomaly detection [11–13] and, lately on, traffic classification [14–20, 27, 28]. It must be said that in this kind of evaluation is not easy to distinguish between the actual impact of sampling from the intrinsic performance issues of the application itself. In this sense, the first part of this work, which studies the effect of sampling on its own, considering different policies, rates as well as a wide range of traffic features well beyond simple volume measurements, is particularly helpful in shedding light on this issue.

As already mentioned, to date there are not many papers jointly considering both traffic classification and sampling [14–20], and, moreover, the majority among them only treats sampling as a minor issue. For instance, [16] analyzes how sampling methodology influences the selection of both elephant and mice flows in the *training* data set, aggravating the traditional class imbalance problem; the same issue is mentioned as particular interesting, but only as a future work in [17]. Other papers only try to predict what sampling might imply for the classifiers they propose: authors of [19], whose technique is based on the size and direction of the very first packets of a flow, sustain that their classifier would badly suffer packet sampling, whereas being robust to flow-sampling; on the opposite side, we argue in [20] that accuracy of stochastic packet inspection should be not influenced by sampling altogether (provided that enough packets are sampled to get statistically relevant signatures).

The impact of packet sampling is experimentally addressed in [14, 15, 18]. In more details, [18] investigates the sampling effect on Reduced Error Pruning Tree (REPTree) classifiers, and limitedly reports a single case study for $p = 1/3$ (asserting that classification accuracy lowers of 10-20%, depending on the client-to-server or server-to-client traffic direction). Instead [14] investigates the sampling effect on a lightweight traffic classification approach (using Naïve Bayes on NetFlow records, and varying the sampling rate) finding that packet sampling does not worsen the results (rather, accuracy may increase under heavy sampling) and suggesting this may be due to an artifact of packet sampling (though a more detailed analysis is missing). Finally, the only work that shares its main focus with ours is [15], which studies the accuracy of statistical traffic classification based on NetFlow sampled data. From extensive experiments and a formal probabilistic analysis,

Table I. Summary of dataset used in this work.

Trace	Auckland	ISP	Campus	UniBS
Year	2001	2006	2008	2009
Packets	291M	44M	17M	26M
Flows	11M	219K	422K	34K
Packets/flow	26.2	202	40.8	764
IPs	410K	61K	81K	6.59K
Available at	[30]	–	–	[31]
Ground truth	Port-based	–	DPI [32]	gt [33]

authors of [15] draw a conclusion similar to ours, showing that the use of sampled data both in the training and testing phase greatly improves the otherwise degraded accuracy obtained with sampled NetFlow. Nevertheless they only consider the limited set of features available in standard NetFlow v5. Finally, both [14, 15] consider only systematic sampling, while in our work we define a biased sampling policy which solves the issue of the number of flows sampled. Additionally, we support our experiments with an analysis of the information content of traffic features, showing that some of them still retain their discriminative power, notwithstanding the sampling degradation.

Building on [29], we extend our previous work by considering a larger dataset, which ensures the statistical relevance of our work, and a larger number of sampling policies, widening the boundary of our investigation. Moreover this work also considers the performance of traffic classification based on sampled data, which was not addressed by our previous work: the most important additions are the analysis of the per-flow feature distortion, the ranking of features according to the their information content and the assessment of classification accuracy.

3. DATASET AND FEATURES

Given the experimental nature of this work, it is extremely important to give as many details as possible on the data employed. In this section we first describe the details of the packet-level traces. Second, we present the tool used to analyze such traffic [21], which applies sampling to the traces and extracts, as well, a wealth of features able to characterize several properties of the traffic at different layers of the networking stack.

3.1. Dataset

In order to gather results that are representative of a wide range of network environments and epochs, we use several traces, whose main characteristics are summarized in Tab. I. In more details:

- **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** is a public available trace [30], collected during 4.5 days in 2001 at the University of Auckland, of which we extract the initial 8hr busy-day period only.
- **UniBS** is a set of 3 traces captured during 3 working-days in 2009 by colleagues at University of Brescia on the 100Mb/s link connecting their campus network to the Internet. This dataset, of which we only use the largest trace, is publicly available in anonymized form [31].

DOI: 10.1002/nem

Table II. Subset of the dataset used for classification, and application breakdown.

	UniBS		Campus		Auckland	
	Flow	Byte	Flow	Byte	Flow	Byte
Protocol	%	%	%	%	%	%
HTTP	49.3	5.6	41.8	62.7	34.8	25.3
HTTPS	1.5	1.2	41.8	30.6	34.8	23.4
FTP	-	-	4.8	0.03	-	-
IMAPS	3.7	0.1	0.2	3.9	0.6	0.9
POP3	1	0.01	-	-	5.6	2.8
SMTP	-	-	-	-	23.9	47.5
Skype	1	0.7	11.1	2.6	-	-
eDonkey	40.1	87.2	-	-	-	-
BitTorrent	3.3	5.0	-	-	-	-

As it can be seen, extremely heterogeneous network scenarios are taken into account in this study: we consider both commercial and academic environments, as well as different access technologies and security settings. Moreover traces are collected in a time which spans nearly a decade, the oldest dating 2001 whereas the newest being from 2009. Such a diverse dataset is fundamental to gather statistically meaningful results. Diversity is even more important for our classification purposes. It is known, in fact, that the accuracy of a classification algorithm is heavily impacted by several factors like different traffic mixes, different network setups, different times of the day and so on. For this reason, all these aspects must be taken into account and possibly included in the dataset, so as to gather a reliable evaluation of the classifier performance. Furthermore, sampling makes this issue more critical, as it possibly discards most of the traffic volume when aggressive rates are applied.

Finally, when testing classification accuracy, finding public traces with reliable ground-truth associated represents a major problem. Were such kind of traces available, they would represent a common ground for researchers to compare the performance of their algorithms and reproduce the results of previous work. Unfortunately, it is well known that this is not an easy problem to solve, for it touches privacy (e.g. private data might be found in packet payloads, hence the need for anonymization) and business interests (e.g. operators are usually very reluctant to share data about their networks). A first step in this direction was taken by our colleagues at the University of Brescia, whose dataset, extensively used in our work, is public and has a very reliable ground-truth associated thanks to their `gt` tool [33]. For the remaining datasets, instead, we need to build our own ground-truth: for Campus we used the DPI classifier described in [32]; for Auckland we employed a simple port-based classification scheme, which was very reliable in 2001, when applications still abide by the standards IANA well-known port allocation; we neglect the ISP trace in the classification part, to avoid using traffic with uncertain application labels.

Tab. II summarizes the composition of the traces according to our pre-labeling. As expected most of the traffic is carried over HTTP, which together with IMAPS, is the only protocol common to all traces. The mix of protocols also reflects the date of the traces: in Auckland we found exclusively traditional client-server applications, whereas more recent traces include also P2P applications, both file-sharing (eDonkey and BitTorrent) and VoIP (Skype).

3.2. Features

In our experiments, packet-level traces were processed with `tstat` [21], which logs several traffic features as output of its analysis. We actually enhanced the tool, adding the possibility of preliminary sampling the input traffic (with configurable policies and rates, as it will be explained later on), before experimentally evaluating the features.

More precisely, `tstat` outputs two different kinds of metrics: some are *per-flow measurements*, i.e. the tools gives the value assumed by the feature for each observed flow; others are *aggregated measurements*, in the form of distribution of the values assumed by the metrics over all observed

Table III. Summary of *aggregated features*, divided by protocol. We report the number of distinct features and, in boldface, the number when considering different traffic direction (i.e., incoming, outgoing, local).

Type	Protocol	Example of features	Number of Features	
			Adirectional	Directional
Single Packet	IP	Packet Length of a IP packet	5	**15**
	UDP	Destination Port of a UDP connection	6	**14**
	TCP	Destination Port of a TCP connection	11	**21**
Multiple Packets	TCP	Maximum RTT of a TCP connection	16	**20**
	RTCP	Average bitrate of a RTCP connection	11	**39**
	RTP	Stream bitrate of a RTP connection	21	**63**
		Total	70	**172**

Table IV. Summary of TCP *per-flow features*, divided by category. Number of features in boldface again includes multiple directions (i.e., client-to-server, server-to-client).

Category	Example of Features	Number of Features	
		Adirectional	Directional
Flow ID	IP addresses of the flow	2	**4**
Flag counts	Number of ACKs sent for the flow duration	5	**10**
Volumes	Number of bytes sent for the flow duration	9	**18**
Packet size	Max segment size of the flow of the flow	3	**6**
Window size	Maximum congestion window size of the flow	9	**18**
Timings	Mean RTT of the flow	7	**14**
Congestion control	RTX timeout of the flow	8	**16**
Flow duration	Completion time of the flow	5	**5**
	Total	48	**91**

flows. Notice that these are just two different points of view of the same measurements (in fact most features are available in either flavor), but they are naturally suited for radically different types of analysis. In this work, we take advantage of either viewpoints, as each of them is best instrumental for one of the two objectives of this work: namely aggregated measurements better reflect monitoring applications, while per-flow measurement are more suited for the classification task.

As it would be cumbersome to report here the full feature list, we refer the reader to [21] for the complete list, and provide here only a few relevant examples for the sake of clarity. Tab. III is a condensed view of the *aggregated features set*, listing the number of features related to different network protocols. As most features are evaluated on different traffic directions we report both the number of distinct adirectional features and the number when considering traffic directions with respect to the measurement point (i.e., incoming to the measurement point, outgoing from the measurement point, local but switched at the measurement point). For what concerns *per-flow features*, we basically concentrate on TCP properties, as they are the most interesting ones for the classification in reason of the protocol breakdown shown early in Tab. II. Tab. IV lists flow-level features divided by type of property that are related to, e.g., traffic volumes, congestion control, timings or TCP flags. Again, the table contains both the number of distinct adirectional features and the number considering traffic directions with respect to the flow initiator (i.e., client-to-server and server-to-client).

We underline that the features we consider are substantially in agreement with the feature set listed in [34], which contains an exhaustive set of features for traffic classification. This agreement follows from the fact that `tstat` started as evolution of Shawn Osterman's `tcptrace` [35], which is also used by authors in [34]. At the same time, the match is not perfect, as e.g., [34] misses some features of `tstat` (e.g. flag stating whether a TCP flow has been interrupted [36], detailed counters about anomalous TCP behavior [37], etc.) and `tstat` does not implement all features listed in [34] (such as the Fast Fourier Transform of the packet inter-arrival time, or the count of valid RTT samples, etc.).

4. METHODOLOGY

We now describe the methodology followed in the experimental study. First, we detail the different *sampling policies* we apply to packet level traces (Sec. 4.1). Second, we present the *statistical metrics* which measure the distortion induced by sampling on feature distributions (Sec. 4.2). Finally, we describe the classification algorithm we employed to test the effect of sampling on traffic classification (Sec. 4.3).

4.1. Sampling Policies

We implemented in the `tstat` tool different sampling policies as defined in [23], and that we overview in the following, explaining their peculiarities with the help of Fig. 1. The picture shows how different sampling policies with the same sampling step $k = 4$ operate on the same sequence of packets; in the picture, packets are represented with different levels of gray associated to different flows, where an "S" denotes a SYN packet.

- **Systematic sampling**: packets are sampled in a deterministic fashion, with 1-out-of-k packets selected. In the example it can be seen that for each 4-packets window, the first packet is always selected.
- **Random sampling**: packets are sampled at random, in particular each packet is sampled independently at a rate $p = 1/k$. As displayed in the example, since the process is completely random, packets might be sampled in sequence, or there may be several consecutive unsampled packets (obviously with a geometrical decreasing probability).
- **Stratified sampling**: k consecutive packets are grouped in a window, in which a single packet is randomly sampled. Looking at the picture, for each 4-packets window, one and only one packet is always selected, but, unlike systematic sampling, instead of selecting always the first, the algorithm randomly chooses which packet to sample out of the four.
- **Systematic SYN sampling**: is the superposition of two independent processes: (i) a systematic sampling process, which selects every k-th packet; (ii) a process which selects all TCP packets with the SYN flag active. This is particular evident in the illustration, where you can see that this policy selects all the SYN packets, in addition to all the packets that a normal systematic sampling would pick.

The first three sampling strategies belong to the family of *unbiased* algorithms, which are the simplest one, being completely unaware of any traffic property. Since these algorithms are extremely lightweight, they are commonly implemented in network equipment, reason why we are particularly interested in their performance. The last one, instead, is what is usually called a *smart* sampling algorithm, because some intelligence is introduced to sample the "right" packets, i.e. the ones conveying the most precious pieces of information. There is no intrinsic limit to the amount of intelligence that one can put in such a sampling method and several smarter algorithms have been proposed by researchers; still, it should be remembered that the purpose of sampling is to reduce computational consumption, thus we want to keep the policy as simple as possible. We argue that Systematic SYN sampling represents a good compromise between these two aspects, particularly for traffic classification. On the one hand, as shown in [38], it improves the estimation of aggregated traffic counters (e.g. total flow length) which are known to be particularly important for traffic classification. Moreover, it ensures that at least one packet for each flows is sampled, or, in other words, that all flows are seen: this solves the problem of results representativeness faced in [14,15] for traffic classification. On the other hand, computational complexity is very low, since the algorithms needs just a counter and a simple check on packet header (furthermore at a fixed offset) to choose whether to sample a packet.

DOI: 10.1002/nem

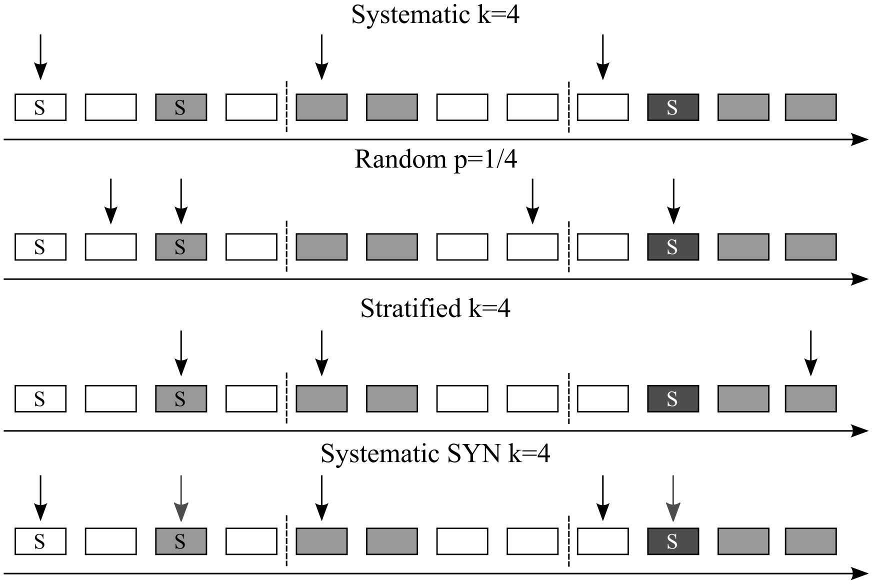


Figure 1. Illustration of sampling policies.

4.2. Metrics

In order to quantify the distortion introduced by any sampling policy into the metrics features by `tstat`, we consider different statistical indexes, suited for either aggregated or per-flow features.

4.2.1. Aggregated features

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:

- **Fleiss Chi-Square (ϕ)**

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

- **Hellinger Distance (HD)**

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

To provide backward compatibility with [3], we consider the ϕ metric, which is a normalized version of the standard Chi-Square metric: increasing values of ϕ correspond to increasing distortion. 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 [3].

The Hellinger Distance (HD) is typically used as a score of similarity between metrics: 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 [39], which also consider other metrics, such as Kullback-Leibler, used e.g., in [40] to reduce the data set size in an approach complementary to sampling.

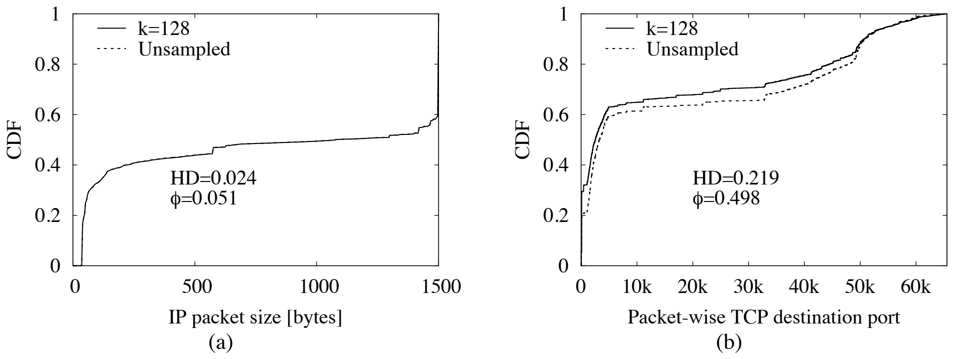


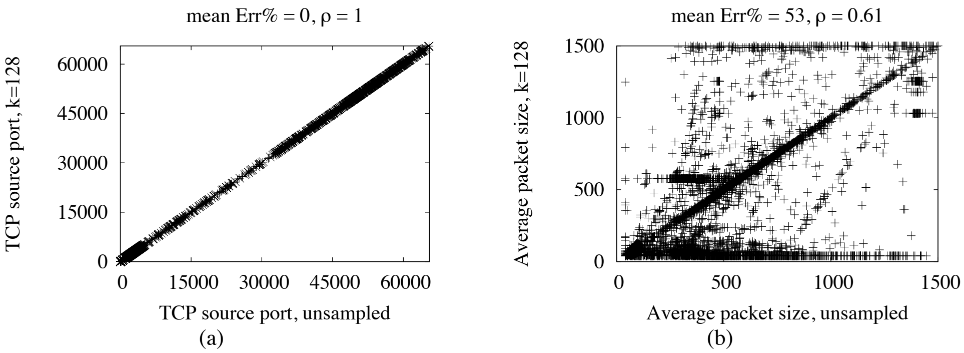
Figure 2. Example of distortion of aggregate features (Campus dataset): CDF of IP packet size (a) and number of packets per destination TCP port (b). Plots report the CDF gathered from the unsampled vs sampled traffic aggregate (systematic sampling, $k = 128$), along with the statistical indexes of distortion.

To have a first idea of the scale of the distortion scores defined so far we provide a preliminary example of some relevant features. Fig. 2-(a) and Fig. 2-(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 for the Campus trace are reported for both original unsampled traffic, as well as for systematic sampled traffic with $k = 128$. Values of distortion metrics are also reported in the picture: for illustration purposes, we select two example aggregated features that are differently impacted by sampling, so to better compare the HD and ϕ distortion score. The CDF of IP packet size (Fig. 2-(a)) 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. Conversely, the packet-wise destination port (Fig. 2-(b)) shows a moderate distortion, with a corresponding degradation of $HD = 0.219$ and $\phi = 0.498$: in this case, differences in the CDF, although of small dimension, can be seen with naked-eyes from the plot. In this latter case, some explanation is necessary. The x-axis represents the TCP port range, so low values of x correspond to packets destined to the well-known port range (i.e., $x < 1024$). Conversely, values of $x > 5000$ correspond to the default Windows ephemeral port range, and $x > 32768$ to the Linux range[†]. Intuitively, as a multitude of clients requests (which use a large range of ephemeral ports) are directed to well-known ports, the difference due to sampling is limited (i.e., since many samples are available over the small range of well-known ports). Server responses coming from well-known ports are instead destined to a large range of ephemeral ports, so that sampling can induce a larger distortion (i.e., as the packet population is more dispersed). Additionally, while requests destined to well-know ports have rather standard lengths (e.g., HTTP GET or FTP RETR commands), the length of the responses may instead widely vary (e.g., due to the varying filesize), causing additional differences. Finally, values of $x > 49152$ correspond to the alternate ephemeral port range, and are apparently relatively less used (i.e., they account less than 10% of the packets).

4.2.2. Per-flow features

Quantifying the distortion in the case of per-flow features is not only useful for monitoring purposes, but also for the classification process. In this case, we compare the exact values measured by our monitoring tool for the same flow with and without sampling. We used two classical metrics to measure the distortion: (i) the mean percentage error *Err%* and (ii) the correlation coefficient ρ between the sampled and unsampled values. The mean percentage error tells us how much the sampled values diverge from the unsampled ones: the smaller the distortion the better; the correlation, instead, tells us whether a linear dependence exists between the unsampled and sampled

[†] Ephemeral port ranges available at `http://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html`



(a) (b)

Figure 3. Example of distortion of per-flow features (Campus dataset): scatter plot of TCP source port (a) and average packet size (b) for unsampled vs sampled traffic, along with statistical indexes of correlation.

values. Like the previous two statistical metrics, the scatter plots of Fig. 3 show two examples of features and the corresponding value of the distortion scores. The right plot is again the average packet size per flows, the same feature whose distribution has been shown in Fig. 2-(a), whereas the left plot shows the source TCP port. The x coordinate is the value of the feature for unsampled traffic, while the y coordinate is the same feature when a systematic sampling with $k = 128$ is applied. As in the previous case, we contrast in Fig. 2 two example features, one of which is not affected by sampling, to better compare the values of the *Err%* and ρ distortion scores.

Two opposite behaviors stand out from the pictures. The feature displayed in the left plot is correctly estimated simply by inspecting a single packet header: for this reason no error is observed and the correlation is maximized. As we will see later on, this kind of features will prove the most valuable discriminators for traffic classification under sampling. On the contrary, the feature in the right plot, being an average, depends on the observation of several packets: therefore, we observe a substantial distortion introduced by sampling, testified both by the large value of relative error, and the lower correlation coefficient as well. In fact, despite many points still align on the $y = x$ bisector line, we can notice a large number of flows falling on a few distinct horizontal lines (namely $y = 40, 576, 1500$). We found that only a single packet was sampled from these flows, which is not representative of the average packet size. In fact, with a single observation, it is likely to get a typical-sized packet (e.g, a $40-$byte packet without 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.

Actually, in the second part of this work, we will be interested more in how sampling affects the relevance of features for the classification, rather than in their mere distortion. Therefore, we need a metric able to capture how much information regarding the application label is conveyed by any given features. For this purpose we resort to the *information gain* [41] metric from information theory. Information gain $I(X, Y)$ measures the reduction of the uncertainty of the class Y (in our case the application label) when you know the value of feature X; in other words, it evaluates how much the knowledge of X tells you about the value of Y. Information gain is based on the concept of *entropy* $H(Y)$ of a random variable Y, i.e., the incertitude of the random variable, which, in case of a discrete random value with a distribution $p(y)$, is given by

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$$

The uncertainty of Y, when the value of X is known, is given by the conditional entropy

$$H(Y|X) = -\sum_{x \in X} p(x) H(Y|X = x)$$

Finally the information gain is the difference of the above quantities

$$I(X,Y) = H(Y) - H(Y|X)$$

The information gain, like entropy itself, is measured in bits of information carried by the feature X. This metric is commonly used for feature selection and in particular we will see in the next session that it is also used by our classification algorithm during the training phase.

4.3. Classification Algorithm

Since sampled traffic has rarely been used for traffic classification, it is still unclear which algorithm might be the most apt to deal with such data. Motivated by work such as [42], which compares different supervised machine learning techniques, and consistently with [15], our choice falls on *Classification Trees (C4.5)*, and more precisely on the open source J48 implementation of the `weka` classification suite [43]. Indeed, as [42] points out, C4.5 offers the best accuracy for a moderate training complexity, with a furthermore very lightweight classification process. Notice however that [42] does not consider Support Vector Machines (SVM), that are well known for their discriminative power, in their evaluation. We point out that while we actually successfully used SVM for the purpose of traffic classification [20,44], for this work we decided to focus on C4.5 – essentially due to the lower complexity of the latter, allowing an extended set of experiments. Notice that any supervised machine learning tool (e.g., SVM, C4.5, etc.) uses as input a *vector* of features and yields a class-label as output of the classification task. Hence, the methodology remains valid to a larger extent of classification techniques (though the performance may change).

As machine learning is out of the scope of this work, we refer the interested reader to [42] for detailed description of different techniques, their merits and performance. At the same time, we need to briefly introduce C4.5, and the notion of information gain, as it will be instrumental to our experimental analysis later on. C4.5 operates on the basis of a decision tree built beforehand from the training points. A new point is classified by traversing the tree from the root to the leaves, choosing the path at each intermediate node (a.k.a. split node) according to the values of specific features: finally a classification outcome univocally corresponds to each leaf. The classification process is extremely lightweight, requiring at most a number of comparisons equal to the maximum depth of the tree. The tree building algorithm is lightweight as well, and it is based on the information gain metric previously introduced. The algorithm basically picks as splitting feature for each node the one which maximizes the information gain: this strategy of using the most helpful attributes at each step is particular efficient, yielding rapidly converging classification trees (i.e., whose depth is limited, requiring thus few comparison per classification operation).

5. AGGREGATE FEATURE DISTORTION

As for the impact of sampling on traffic measurement, in this section we concentrate on the pure distortion introduced by sampling observing the impact it has on aggregate traffic metrics. We first characterize the overall feature set and the range of value scored by our statistical indexes, in order to have a general picture (Sec. 5.1). We then focus on smaller sets of features defined by the protocol layer they pertain to (e.g. network or transport layer), identifying which family of attributes is more heavily affected by sampling (Sec. 5.2). We finally individuate a set of robust features, whose distortion keeps bounded even under heavy sampling, on which we investigate the impact of different sampling policies (Sec. 5.3).

5.1. Overview of Sampling Impact

In this first part we look at the complete set of features at once, in order to observe the general trend of feature degradation under sampling and to better understand the range of variation of the statistical metrics. For this purpose we use the distributions of the two distortion scores over the whole set of features reported in Fig. 4 and represented in the form of complementary cumulative

DOI: 10.1002/nem

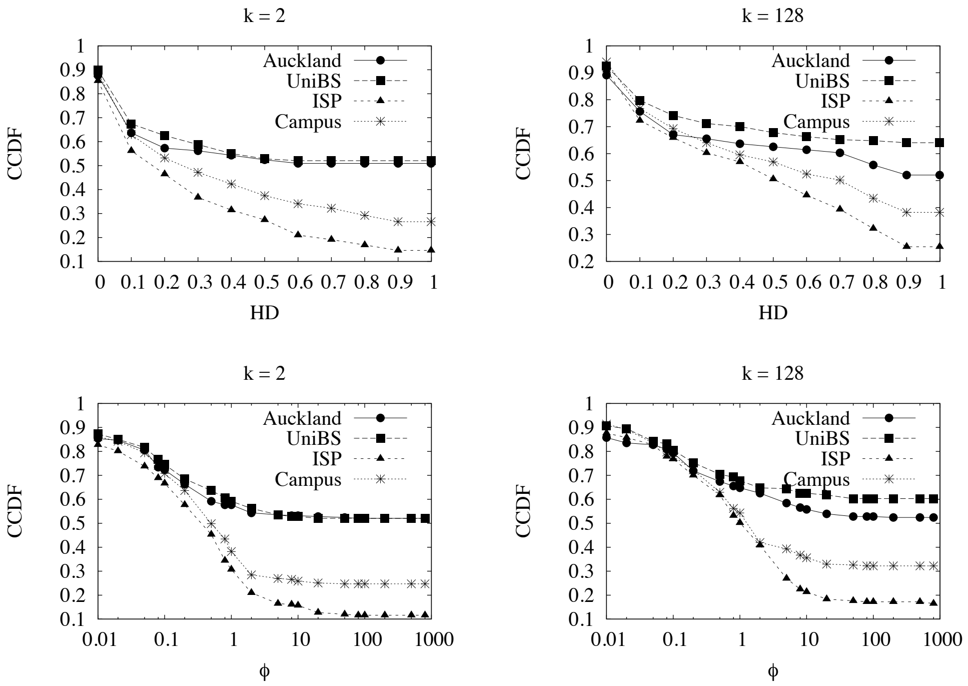


Figure 4. Distribution of the Hellinger distance and ϕ coefficient over the whole features set for systematic sampling and $k \in \{2, 128\}$.

distribution function (CCDF). The graphs report the distributions for systematic sampling with $k = 2$ and $k = 128$, respectively in the left and right column, while the top row is related to the HD score and the bottom one to the ϕ coefficient. We keep a separate curve for each dataset as they represent different network conditions and traffic mixture, thus we prefer to observe their behavior individually.

At first glance, the two statistical metrics are quite coherent with each other, showing practically the same trends, even though HD is bounded in the interval $[0, 1]$ whereas ϕ is not. Moreover, it looks as Campus and ISP appear more robust than the other traces, as features degrade more gently: the heterogeneous traffic mixture found in this traces is likely the key reason why they are less affected by sampling.

On the contrary, it is quite striking that for the other traces around 60% of features are completely distorted (i.e., the distance metric is maximized) already with $k = 2$: this means that they are impossible to evaluate even with very light sampling. On the other hand, 10% of features show no distortion at all, scoring a zero for both metrics, meaning that they are perfectly estimated regardless of the sampling rate. As a matter of fact, all these features are not distorted as they can be correctly measured simply by inspecting one single packet of any given flow.

To dig further, we look at the same data in a different way by means of the scatter plots of Fig. 5, where each feature is represented with a point whose x and y coordinates are respectively the distortion score for $k = 2$ and $k = 128$. We have zoomed to show the area close to the origin, where we see a cluster of points showing no degradation.

However, the pictures show an interesting *artifact*: notice that the estimation of a few features *apparently* seems to improve under heavy sampling, as shown by the points falling in the gray-shaded part of the graphs (some of which are labeled with the feature name). This weird effect is mostly due the different way sampling impacts on long and short flows, since long ones have a larger probabilities of being sampled while the short ones are likely to disappear after sampling. For instance, this effect is particularly evident for the `tcp_cl_b_l_c2s` feature, i.e. the distribution

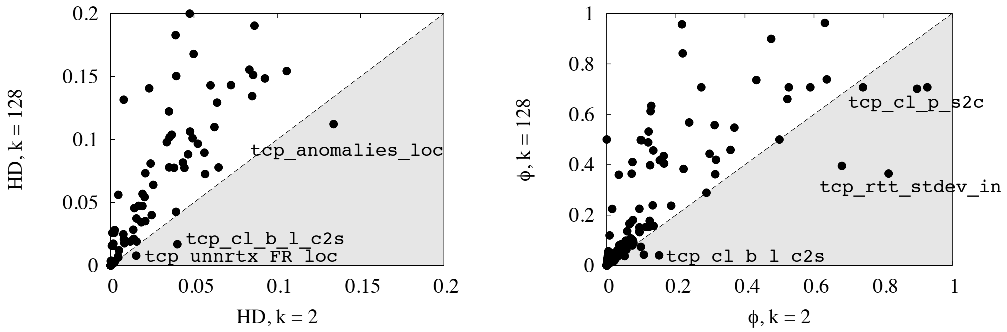


Figure 5. Scatter plot of Hellinger distance and ϕ coefficient over the whole features set for systematic sampling with $k \in \{2, 128\}$.

of the TCP flow length, measured with a coarse granularity (i.e., large bins). 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 bins. 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 finer granularity `tcp_cl_b_s_c2s`, since in this case it is less likely for the sampled feature to fall into the same bin of the unsampled traffic). The community should be aware of these artifacts that, if neglected, could undermine the results of an experimental investigation.

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 more points in this area. Moreover for the ϕ coefficient many features actually fall closer to the bisector as well, which means that only a slight degradation is detected in spite of an 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 larger distortion score in ϕ. The main outcome of this behavior 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.

5.2. Impact of Protocol Layer

We now group the features in different subsets according to the protocol layer: in particular we consider IP features, UDP single-segment features, TCP single- and multiple- segment features as in Tab. III. 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.

In the light of what observed in the previous section, without loss of generality nor of information, we express the distortion scores using the Hellinger Distance alone. For the time being, we still focus on a single sampling policy (namely, systematic sampling), delaying the consideration of different sampling policies to the following section. However, we do take into account a large range of sampling rates, from 1/2 to 1/1024. Results are reported in the four graphs of Fig. 6, corresponding 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.

A general observation which holds for all of the datasets, is that some features prove to be intrinsically easier to measure under sampling. For instance the curves of distortion scores for both IP and UDP single-segment features are considerably closer to the minimum value for the HD across

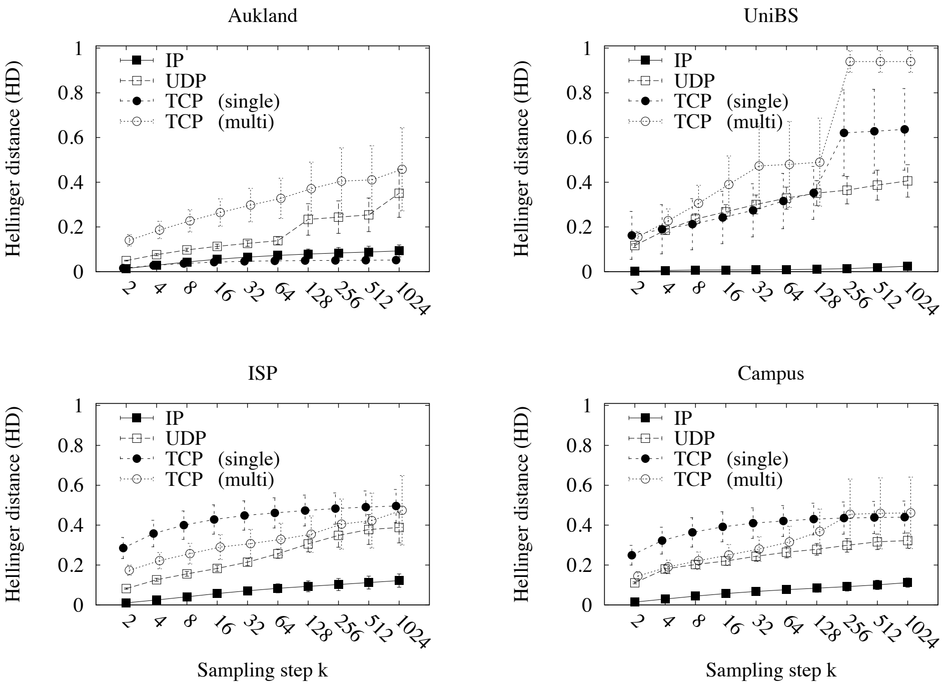


Figure 6. Mean and variance of the HD distortion score, for features grouped by protocol layer, as a function of the sampling step under uniform sampling policy. The extremely low distortion of "TCP single" features in the Auckland trace is an artifact due to the lack of packet payload, in particular of TCP options.

all dataset (apart from Auckland, as we will explain later). This confirms our previous intuition that features relying only on the inspection of a single packet (e.g., IP packet size) are more robust to sampling than features depending on the observation of multiple packets (e.g., RTT time).

On the contrary, while ISP and Campus exhibit rather consistent and coherent trends, Auckland and UniBS have some anomalies. For instance, in the Auckland trace, the single-segment TCP features have an unusually low distortion score. Investigating this issue further, we found that in this dataset TCP options (e.g. MSS negotiation, window scale) are obfuscated for privacy reasons (actually set to 0) together with the rest of packet payload, which makes `tstat` unable of correctly estimating 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 the impossibility of correctly estimating most of the features of that group from the trace under investigation, even in the unsampled case. Instead, the higher degradation detected for TCP features in the UniBS trace derives from the artificial nature of this trace, built with a few hosts automatically generating specific traffic. As a result, the trace includes many elephant flows (notice the large mean flow size in Tab. I) and when sampling is applied this strongly biases the distributions, yielding larger distortions.

Campus and ISP instead, show a similar behavior when considering TCP features as well. Interestingly, notice that, at lower sampling rates, TCP features depending on multi-segment suffer a smaller distortion than 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 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 be safely estimated for low sampling steps, as all segments anyway carry useful information that can improve the feature estimate.

5.3. *Impact of Sampling Policy*

In this section, we assess the impact of different sampling policies on the accuracy of traffic feature estimation. However, first we need to define the subset of features to use for our analysis: this is not an easy choice given the large number of features already distorted for small sampling steps on the one hand, and the extremely varied behavior of different feature groups across each trace. Therefore, we adopt a simple but effective threshold-based selection criterion: we consider as robust, and focus on in this section, all features whose HD distance is lower than a predefined threshold. Hence, 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.

Without loss of generality, results in this section refer to features which have an $HD < 0.3$ with a sampling of $k = 128$. Notice that we select this values of threshold in reason of the knee of the distribution observed in Fig. 4. Notice also that different threshold values, as the $HD < 0.1$ we used in [29], yield to similar considerations: yet, as in this work we consider a larger dataset, and we require features to be robust in all datasets, we prefer to apply a less stringent threshold, so to assess the impact of sampling policy on a larger number of features.

The final set contains 36 features, equally distributed over the 3 protocols IP, TCP and UDP. Thus, each protocol layer is represented in the robust set, except for the RTCP and RTP layers. In fact, the relatively low amount of RTP/RTCP traffic present in the Auckland dataset makes it difficult to evaluate the related features for this trace, especially when hard sampling conditions further limit the number of valid samples.

Results of this analysis for the robust features set are reported in Fig. 7, composed of one graph 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 four curves, one for each dataset, depicting the average distance score over the robust features set; variance of the distance score is also reported by means of vertical error bars (notice that we employ variance instead of standard deviation, as the latter is visually noisy, as the square root of HD values in $[0, 1] \in \mathbb{R}$ explodes).

At first glance, we can observe that there is no clear advantage in the choice of random sampling or systematic sampling: considering the corresponding 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 [3]. 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 [8], which however limitedly considers only traffic volume measurements under sampling (i.e., flow length).

Conversely, our smart sampling policy has a noticeable impact on measurements accuracy, yielding completely different trends in addition to high distortion scores for high sampling steps. Intuitively, Systematic SYN sampling heavily biases the distribution, as by definition it samples at least one packet for each flow. At high sampling steps, the SYN packet is likely to be also the only sampled packet, which introduces a significant distortion in the aggregate features. In other words, there will be a large portion of flows with a single sampled packet, and therefore rather poor estimation of all relevant flow properties. While, as we will see, traffic classification accuracy is not affected by this distortion, for the time being we can conclude that biased techniques are not indicated in the context of traffic monitoring and characterization.

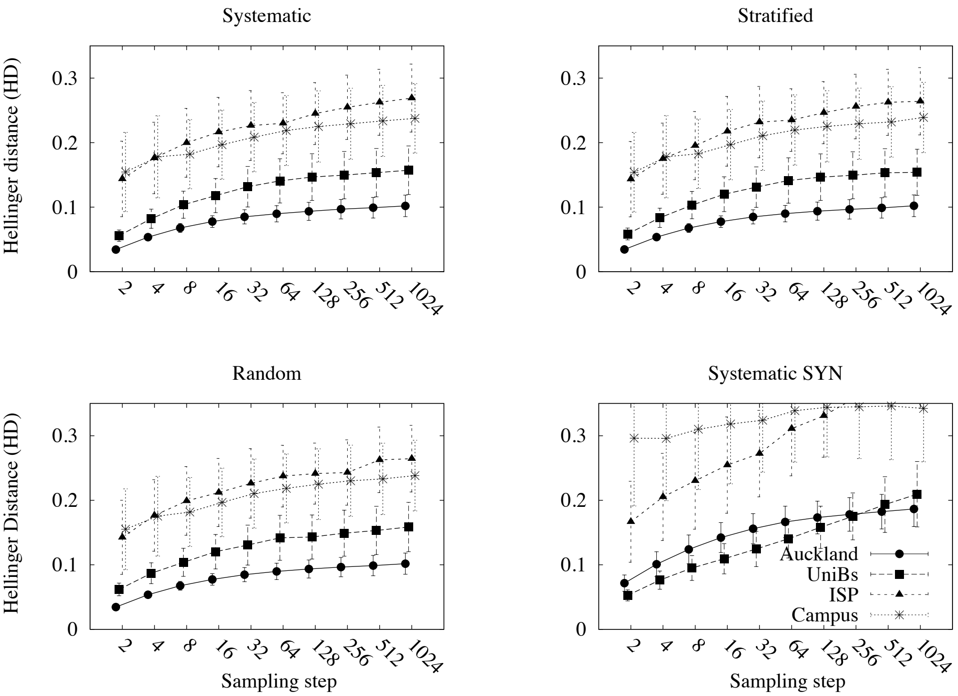


Figure 7. Mean and variance of the HD distortion score for the robust group of features as a function of the sampling step for different sampling policies.

If we focus on the difference between datasets, we see that they are ranked almost in the same order by different sampling policies: apparently Auckland and UniBS are the easiest one, which seems to be in contrast with our previous analysis. Yet, recall that we have conditioned distortion scores over the robust subset of features: in the case of Auckland, this means that we are averaging with TCP features getting extremely low scores due to the measurement artifact early outlined; in the case of UniBS we have removed exactly those features that we saw heavily distorted in Fig. 6.

It is also worth noticing that sampling error saturates, in the sense that distortion scores do not increase as fast as the sampling step, which is exponential. The reason of such a behavior is twofold: first, as most features are estimated from the observation of a single packet, they degrade gently when increasing the sampling step; second, some of the artifacts showed in the previous section may still arise (i.e., features whose distortion score decreases rather than increasing for higher sampling rates).

6. SINGLE-FLOW FEATURE DISTORTION

To deepen the analysis of the impact of sampling on traffic measurement, in this section we study the distortion of per-flow features and, with respect to the former section, we change both the viewpoint and the metrics employed to measure the impact of sampling. We also apply the information gain metric to assess the amount of information conveyed by each feature: such analysis is instrumental for the evaluation of traffic classification accuracy, that we will carry on in the next section.

To avoid cluttering the pictures, this preliminary analysis will be done considering only the UniBS dataset, whose ground-truth is the most reliable; however, we will come back to the whole dataset for our last experiments to draw our general conclusions.

In the remainder of this section we always refer to Systematic SYN sampling, which was introduced specifically for traffic classification, as it gives us two main advantages over the other

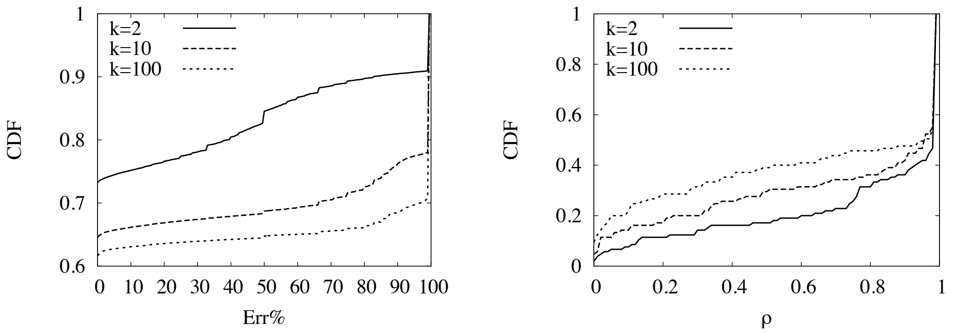


Figure 8. CDF of (left) Err% and (right) ρ for UniBS trace and different sampling step.

policies. First it overcomes the problem of dataset representativeness, for at least one packet per flow is always sampled: in this way even protocols with short flows (corresponding to a small probability of being sampled) are, nevertheless, included in the dataset after sampling. Second, the SYN packet carries very important information about the related flow (e.g. initial sequence number, initial timestamp, eventually some TCP options) which `tstat` can leverage to improve the estimation of many flow properties.

6.1. Overview of Sampling Impact

Following the same approach of Sec. 5, we first consider the whole set of per-flow features, to gather an overall picture of the impact of sampling. Fig. 8 shows the distributions of the relative percentage error and the correlation coefficient between the sampled and unsampled values of all features for the UniBS trace, comparing different sampling steps represented with distinct curves.

Focusing on the percentage error plot on the left, a large number of measures appears not affected by sampling, scoring a 0% error; we must not forget, however, that the distribution may be biased towards zero by features that are estimated by a single packet inspection, or that score their default value that `tstat` assigns to features it cannot evaluate. On the other hand, an increasing portion of the features, from 10% for $k = 2$, to 30% for $k = 100$, are completely distorted with an error greater than 100%. Interestingly, this partition of the features becomes more and more sharp with increasing sampling step. However, high distortion does not necessarily imply that such features is useless for traffic classification: indeed, provided that distorted features are still clearly *separable* across applications, their information would still be extremely valuable for classification purposes. For completeness sake, we reported also the correlation coefficient distributions in the right plot, from which the same conclusion can be drawn the same percentage of features that scored a 0 error gets the maximum value of correlation, and again higher sampling steps cause a larger degradation highlighted by a larger portion of features with small correlation with the unsampled case.

Let us now focus on specific features. We present two examples in the scatter plots of Fig. 9: the left plots are related to the maximum packet size observed in a flow, while the right ones show the average RTT[‡]. As previously done, we select two rather different features: one has a low distortion (maximum packet size) and we expect it to carry some important information for traffic classification; the other (RTT) is instead affected by sampling and the geographical host distribution, but is otherwise unrelated to traffic type. On the x-axis we report the value of the feature in the absence of sampling, while on y-axis the one when a sampling with $k = 100$ is applied. We represent flows belonging to traditional client-server (CS) applications in the top plots and to

[‡] We relax the RTT computation by avoiding to match TCP sequence numbers, and simply take the time elapsed between consecutive packets in different directions. Notice that sampled RTT is roughly two orders of magnitude bigger ($k = 100$) with respect to the unsampled one: hence, in principle we could correct the sampled RTT by scaling it by a factor k. At the same time, since the RTT feature is unrelated to the traffic type, and thus not relevant for the purpose of traffic classification, we avoid to introduce the correction

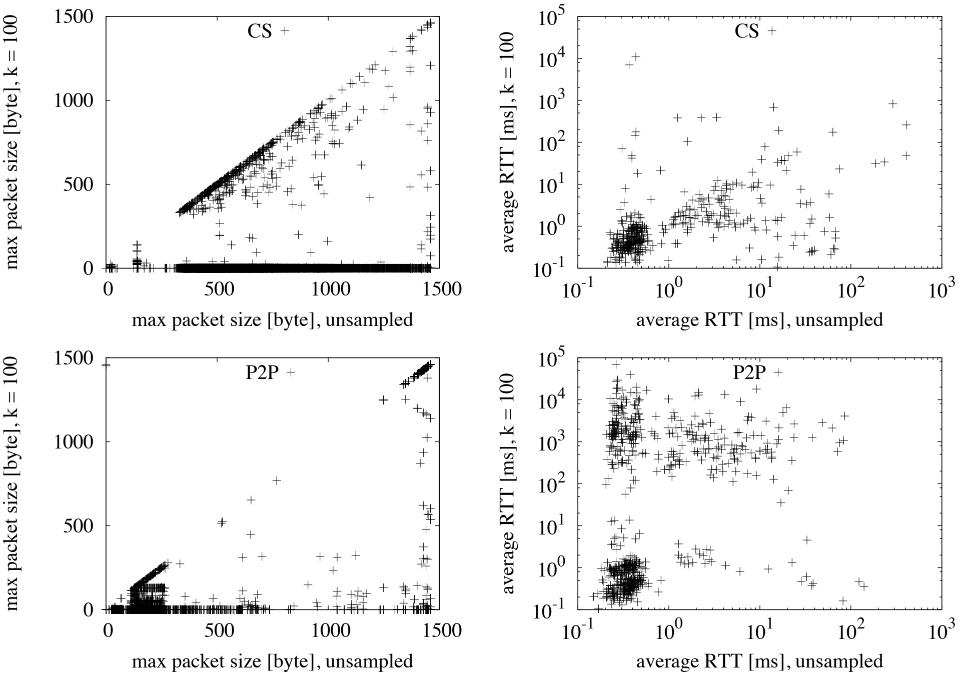


Figure 9. Scatter plot of features values for unsampled and SYN Sampling $k = 100$ for UniBS trace, contrasting peer-to-peer (P2P) and traditional client-server (CS) applications.

peer-to-peer (P2P) applications in the bottom plots, in order to see whether different classes of applications correspond to different behaviors under sampling.

The plot allows us to gather some observations. For the packet size, while P2P applications employ either very small (signalling) or full-size (data) packets, client-server applications often use medium-sized packets as well. Yet, for CS applications the features are underestimated mostly of the time, meaning that usually only a single small packet (e.g., SYN) of a short CS flow is sampled due to biased-SYN sampling; on the other hand, this same feature is correctly evaluated for P2P applications, whose longer exchanges increase the odds that the monitor tool samples bigger packets as well. In the right plots, we can find again two clouds: the top one, concentrated specially in the top left corner for P2P, and the bottom one, concentrated in the area close to the origin, which is equally composed by P2P and CS flows. A possible explanation of this behavior is that many P2P flows have intermittent but long-lived flows (for instance because they speak with a recurring peer after a while), which yields estimation of very large RTTs (due to pairing the ACK with the wrong data message); on the contrary, CS applications have shorter flows, that reduce the likelihood of pairing a very late ACK and allow an easier measure of the RTT. Again it looks like traffic belonging to different applications is differently impacted by sampling, which is good news for our classification purposes.

6.2. *Ranking Features*

After this preliminary evaluation of feature distortion, we want to assess the degradation of the *amount of information* conveyed by features due to traffic sampling. This analysis is complementary to the one presented in [15], where authors analytically study the error introduced by sampling in NetFlow features, but do not evaluate whether sampling also impacts the *information* content of such features, which is a critical component of the classification process.

To measure this quantity we finally use the Information gain score introduced in Sec. 4.2.2. We use this measure to partition the features set in two groups, using a simple threshold-based criterion: (i) the *most-relevant* feature group including features whose information gain is greater than 1 bit

Table V. Feature Information gain for UniBS trace at different sampling rates.

Features	**Unsampled**		**Sampled k=2**		**Sampled k=10**	
	Score	Rank	Score	Rank	Score	Rank
Server-IP-address	**1.68**	1	**1.68**	1	**1.68**	1
cwin-min-c2s	**1.49**	2	**1.20**	6	0.60	14
min-seg-size-c2s	**1.48**	3	**1.22**	5	0.47	23
cwin-max-c2s	**1.47**	4	**1.11**	8	0.56	15
max-seg-size-c2s	**1.43**	5	**1.17**	7	0.46	24
initial-cwin-c2s	**1.41**	6	0.71	26	0.29	32
First-time	**1.37**	7	**1.37**	2	**1.37**	2
cwin-min-s2c	**1.35**	8	**1.06**	11	0.53	16
Server-TCP-port	**1.34**	9	**1.34**	3	**1.34**	3
initial-cwin-s2c	**1.33**	10	0.77	22	0.30	31
Client-IP-address	**1.31**	11	**1.31**	4	**1.31**	4
cwin-max-s2c	**1.28**	12	0.99	14	0.49	21
min-seg-size-s2c	**1.22**	13	0.96	16	0.51	19
max-seg-size-s2c	**1.21**	14	**1.03**	12	0.50	20
Last-time	**1.14**	15	**1.09**	9	**1.02**	5
win-max-s2c	**1.08**	16	**1.07**	10	0.98	6
Completion-time	**1.03**	17	0.97	15	0.42	25
win-min-s2c	**1.02**	18	**1.01**	13	0.94	7
unique-byte-s2c	**1.02**	19	0.74	23	0.42	27
data-byte-s2c	**1.01**	20	0.74	24	0.42	26

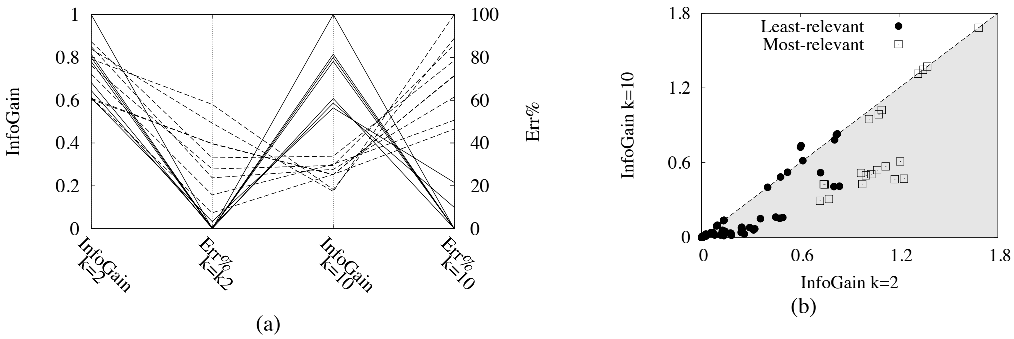
(i.e. that are able to discriminate between two labels); (ii) the least-relevant feature group, containing all the remaining features. The performance of such subsets of features will be evaluated in the next section; here we report the scores for the most-relevant features in Tab. V together with their rank, comparing the unsampled case with the sampled case with $k \in \{2, 10\}$.

The information-gain metric partitions our feature set as expected. For instance, features like the client ephemeral source port, which is chosen randomly upon connection setup, clearly ends up in the least relevant feature set, being useless for the classification process. On the other hand, the server IP address exhibits always a high score regardless of the sampling rate: both the fact that this feature is correctly estimated by inspecting a single packet and the reduced number of servers in the UniBS trace concur to make this value a strong discriminator.

Besides, notice that the larger the sampling period, the smaller the number of features showing scores greater than 1 bit. Additionally, since the ranking changes from one sampling rate to the other, this suggest that the most-relevant feature set gathered for unsampled traffic may no longer be the same for higher sampling.

To visually represent the effect of sampling on the most-relevant feature set, we use the parallel coordinate plot of Fig. 10-(a). Each line represent one feature and connects the Information gain score and mean percentage error for that metric for the two sampling steps $k \in \{2, 10\}$. Two evident patterns emerge from the picture, which are highlighted by means of two different line types. Full lines clearly represent those features evaluated from a single-packet inspection: they have high information-gain, which means they are extremely correlated with the application label, along with a low relative error distortion, which means they are correctly measured regardless of the sampling step. On the other hand, features denoted by dashed lines, though more degraded by sampling (their percentage error increases with sampling), are included in the most-relevant set because they still bring benefit to the classification, as testified by high information gain for $k = 2$, moreover only moderately decreasing for $k = 10$. Hence, while these features are degraded under sampling, they still allow to separate application labels as exemplified in Fig. 9

In Fig. 10-(b) we extend our considerations to the whole set of features: each feature is represented by the point whose coordinates are the information gain score for $k \in \{2, 10\}$ respectively on the x and y axis, using different point types for most relevant (empty squares) and least-relevant (filled circles) features. First, the picture confirms that the ranking of features is not stable: notice that the score of some least-relevant features exceeds the one of a few most-relevant ones for $k = 2$ (remember that the partition of the feature space has been performed using the ranking of unsampled



(a) (b)

Figure 10. (a) Parallel coordinates plot for most-relevant features and (b) scatter plot of information gain for all features with $k = 2, 10$.

features). This behavior is more evident considering $k = 10$: some features in the least-relevant group should be considered as good discriminators, not only because of the higher scores, but rather because they fall on the bisector, showing no degradation of their information content. Overall, it looks like it is extremely difficult to predict how the information content of a feature might be degraded by sampling, as this may furthermore vary depending on the sampling step. This suggests using the whole features set and letting the classification algorithm deal with that. In the next section, which compares the classification accuracy of different features sets, we will see whether this is a good strategy.

7. TRAFFIC CLASSIFICATION UNDER SAMPLING

After the analysis of the distortion due to sampling in both aggregate (Sec. 5) and single flows features (Sec. 6), in this section we provide a detailed evaluation of the impact of the sampling on traffic classification. More precisely, we first gather baseline performance for different features sets for unsampled traffic (Sec. 7.1) and then we measure the impact of training policies (Sec. 7.2) for sampled traffic, considering the UniBS trace. Finally, we extend our investigation to other datasets as well (Sec. 7.3). We report the overall accuracy of the classification and omit the detailed per-application accuracy as such an analysis is already found [15] and we had rather concentrate on uncovered aspects of traffic classification under sampling.

7.1. Impact of Feature Set

We start by comparing the classification performance of different sets of features with unsampled traffic, using only the UniBS trace. We consider the following sets:

S1 **baseline-features** is a simple set of features, that basically contains the information derived by a flow-level monitor (e.g. NetFlow) defined as in [14].

S2 **all-features** is the whole set of features produced by `tstat`, coherent with [34].

S3 **no-IPs**, obtained removing from the whole set both source and destination IP addresses, i.e. $S3 = S2 \setminus \{srcIP, dstIP\}$.

S4 **no-IPs/Time/Flags**, obtained removing from the whole set IP addresses, TCP timestamps and flags, i.e. $S4 = S2 \setminus \{srcIP, dstIP, timestamp, flag\}$.

S5 **no-Ports**, obtained removing from the whole set both source and destination transport layer port, i.e. $S5 = S2 \setminus \{srcPort, dstPort\}$

S6 **no-IDs**, obtained removing from the whole all flow identifier $S6 = S2 \setminus \{srcIP, dstIP, srcPort, dstPort\}$

S7 **most-relevant**, comprising the features listed in Tab. V, formally defined as $S7 = \{x \in S2 : InfoGain(x) > 1\}$

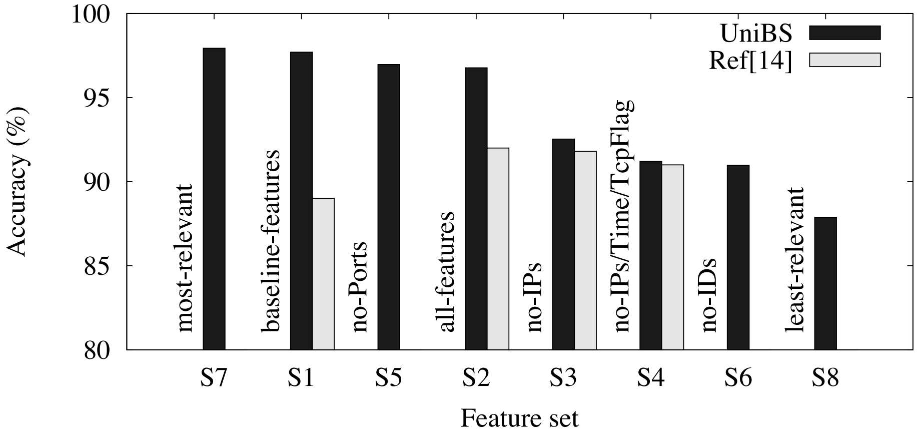


Figure 11. Classification accuracy achieved by different features sets with unsampled traffic for UniBS trace.

S8 **least-relevant** complementary of the most-relevant features set, i.e. $S8 = S2 \setminus S7$

The first four groups are in common with [14], in order to have a direct comparison with previous work. We notice that features included in the baseline-features set *S1*, such as destination IP and port, start and end time and total transferred bytes, were already identified as the most reliable ones, showing high information gain in our earlier analysis (Sec. 6).

The accuracy achieved by these features sets in our experiments is depicted in Fig. 11, where we also plotted the results reported by [14], where available, as a comparison. Yet we underline that this is an *indicative* comparison as results are not directly comparable for three reasons: first, the dataset differs; second, the machine learning technique differs; third, there are slight differences in some feature sets as the overall set $S2$ produced by `tstat` is a super-set of the one considered in [14] (see the discussion in Sec. 3.2).

Speaking about the comparison, we gather that for the features sets $S1$ and $S2$ the accuracy for the UniBS dataset is higher than that reported in [14], whereas values scored by sets $S3$ and $S4$ are coherent with previous results. To explain this behavior, we must go back to the information gain score. We have noticed before the high correlation between the application label and the network-layer identifiers for the UniBS trace (i.e., IP address of the server), which causes the performance drop from sets $S1$ and $S2$ to sets $S3$ and $S4$.

Considering all subsets, we gather that, as expected, the most-relevant feature set exhibits the best accuracy, though being the smallest one. The complete set $S2$ instead turns out in a slightly worse result: we see a little overfitting phenomenon (i.e. useless features disturbing the classification process), but the difference is negligible in our case. Therefore, we estimate the prominence of feature selection less relevant, and will consider other, less explored, issues in what follows.

We make a few final remarks before changing subject. The performance of sets $S3, S5, S6$ further shows that IPs are much more relevant than ports, causing a larger decrease in performance when removed from the feature set. Finally the last vertical bar, though referring to a quite numerous set of 85 features, shows as expected the worst performance as it comprises features carrying less information about the application label. Notice that accuracy however exceeds 85%, meaning that the set of the least-relevant features still includes valid discriminators.

7.2. *Impact of Training Policy*

Clearly, the selection of flows to include in the training set has a great influence on the final classification accuracy (i.e., how to find representative samples, how to face the class imbalance problem, etc.). Yet, in this paper we are more concerned in a novel factor: i.e., whether training and validation data should be gathered at the same sampling rate.

DOI: 10.1002/nem

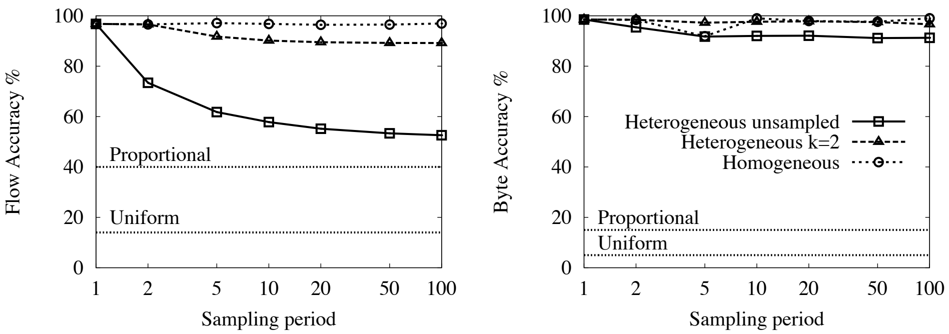


Figure 12. Impact of Homogeneous vs Heterogeneous training set policies at varying sampling rates in terms of flow and byte accuracy.

This is an important point as it means that ISP may need to use different classification models, one per each sampling rate, or may use a single unique model for all rates. In the following, we use k_T to denote the sampling step used for the training data, and k_V for the one used for validation data.

- **Homogeneous classification**, in which training and validation sets contain data obtained with the same sampling rate (i.e., $k_T = k_V$). This corresponds to the case of ISPs using different training sets, one for each sampling rate. Results shown in the previous sections were gathered using this training policy.
- **Heterogeneous classification**, in which training and validation sets contain data obtained with different sampling rates (i.e., $k_T \neq k_V$). Intuitively one may think that richer data with a lower sampling period might contain more information that could be successfully exploited for the classification. At the same time, we may expect that under sampling, the feature estimation error will grow large, with a corresponding information loss. In our experiments we investigated the full space resulting from the cross product of $k_V \times k_T$, but we report here only two examples: first the extreme case where we train the machine with unsampled data ($k_T = 1$) and then the results gather with $k_T = 2$, i.e., the minimum level of sampling.

We test these policies on the UniBS trace, performing a cross-validation and using 10% of this data as training set and the rest as validation set. The cross-validation procedure repeat the train/validation process 10 times, randomizing each time the training set (and changing the validation set as a consequence). As for the class imbalance, we took extra care in building the training set so that the proportion of the different application are the same of the original data (i.e., as 49% of the original trace is constituted by HTTP flow, the training set is composed for 49% of HTTP flow samples).

In Fig. 12 we report the flow (left plot) and byte (right plot) accuracy obtained by the C4.5 algorithm with the complete set $S2$ of features provided by `tstat` (whose performance was only slightly affected by overfitting with respect to the most-relevant feature set), for both the heterogeneous and homogeneous cases, for increasing sampling step. We report two cases of heterogeneous policies: first the case where the classifier is trained with unsampled data, i.e., $k_T = 1$; second the case where the classifier is trained with lightly sample data with $k_T = 2$. As a reference and lower bound, we also plot the results of two dummy classification processes: (i) *Uniform* selects the classification label uniformly at random among the possible classes; (ii) *Proportional* selects the label at random, but with a probability proportional to the number of flows belonging to that class.

Looking at the flow accuracy, the homogeneous case exhibits the best results, achieving an high accuracy which furthermore does not deteriorate under more aggressive sampling. In the heterogeneous unsampled case, the accuracy drops considerably already with a sampling period of $k_V = 2$ and then decreases until at $k_V = 100$ it achieves only slightly more than 50%, which is close

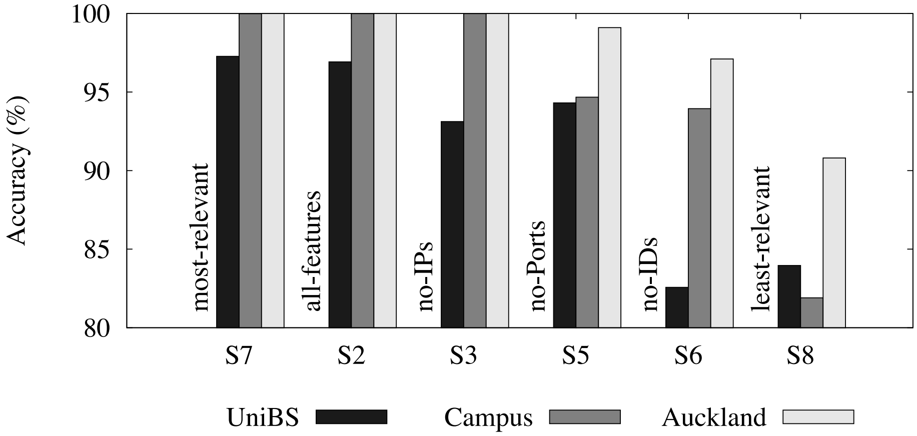


Figure 13. Flow classification accuracy for different traces and features sets (homogeneous training, $k = 100$ sampling).

to the proportional random labeling process: in other words, the heterogeneous training process was only able to correctly learn the *flow proportion* (due to our balanced training set selection policy). The heterogeneous case with $k_T = 2$ has a better performance, though far from the homogeneous case: this means that, whatever the sampling step k_V, it is always better to train the classifier with already sampled data (i.e., $k_T > 1$) when this is the kind of data to be classified. Interestingly, there is a much smaller difference between the two training policies in term of byte accuracy even in harsh sample conditions: this means that elephant flows are always correctly classified, whereas classification errors are much more frequent for mice flows.

In agreement with [15], these results show that, even though features are distorted, the amount of information they convey on the application label is still relevant for the classification, as shown by the analysis in the previous section as well. On the contrary, while unsampled data well captures the properties of real traffic, it is unsuitable to characterize, and therefore classify, the sampled traffic. Hence, ISPs shall use a specific classification model for each sampling rate. In case of variable sampling rates, this may not be feasible: however, as classification degrades smoothly between the homogeneous $k_T = k_V$ and heterogeneous $k_T = 1$ cases (i.e., k_T=10 performs better than k_T=1 for k_V=100), a viable compromise would be to select the closest k_T from a set of models.

7.3. Impact of Dataset

Finally, we extend our analysis to the other traces. In principle we were extremely interested in testing the *portability* of the method: i.e., to assess the performance of a model trained on a trace and validated on the other traces. Unfortunately though, due to the heterogeneity of the dataset, a thorough and coherent comparison is possible only with an extremely reduced set of protocols (only HTTP, HTTPS and IMAPS are common to the three traces, and no more than 4 protocols are common to any 2 traces), resulting in a statistically not significant comparison. Therefore, we limitedly report results considering each dataset in isolation[§]. We use a sampling period of k=100 (a common value used in operational networks), the C4.5 algorithm in the homogeneous case, with data split again in 10% and 90% for training and validation sets respectively. The whole classification process is performed separately for each dataset, using the sets of features defined in Sec. 7.1, and results are reported in Fig. 13. Notice that, as the Auckland ground truth is established

[§] We however performed the portability experiments (i.e., testing on each trace and validating over the others) on the reduced protocol space, obtaining rather low accuracy performance. To give the reader the intuition why this happens, consider that IP address is the most prominent discriminator, and that IP addresses clearly differ across traces.

based on well-known ports, we expect 100% accuracy when this feature is included in the feature set (on which the ground truth for this dataset relies).

The set of most-relevant features is consistently the one yielding the best result, for all datasets. Interestingly, also the set composed by all features achieves nearly the same performance, in contrast with the common belief that classifiers are misled by an excess of information. As a matter of fact, notice that the set of least-relevant features still contains a valuable amount of information as it correctly identifies more than 80% of flows for any trace.

As previously done, we also investigate the importance of IP addresses and transport-layer ports features by evaluating the classification accuracy after their removal. We find that transport layer identifiers are particularly important for classification purposes, hence removing them has a great impact on the accuracy (especially for Auckland as expected). Conversely, IP addresses are particularly relevant only for the UniBS trace, where, as previously observed, the same server IP is very often associated with the same application, thus becoming a good discriminator; while in more diverse traces removing IP addresses from the feature set does not affect the performance at all. Removing both transport and network layer identifiers gives the worst results – for the UniBS even worse than the least-relevant set of features. Moreover, the performance loss may be more pronounced than in the unsampled case shown earlier, meaning that information contained in IP address and ports is even more important under sampling. At the same time though, the removal of IP and ports does not drastically compromise the accuracy in the Auckland case, where the features of the different traffic classes are likely more separated.

8. CONCLUSION

In this paper we empirically studied the impact of packet sampling on traffic measurement and traffic classification. Sampling is already a very common practice in operational networks and the increasing trend of network traffic is likely to spread its adoption even more among operators. For this reason, in this work we accurately assessed the amount of information lost when applying sampling to traffic characterization, as well as the repercussion of such loss on the performance of different applications of sampling data, in particular of traffic classification.

We processed an extremely heterogeneous dataset composed of four packet traces (representative of different access technologies and operational environments) with a traffic monitoring tool able to extract several traffic features both in aggregated and per-flow fashion. The tool was modified ad hoc to apply different sampling policies and arbitrary sampling rates to the traces. Moreover, in an attempt to foster cross-comparison in the community, we made an effort with respect to both previous literature (i.e., by considering the same features sets of [14]) and future research (i.e., by using open datasets and describing our labeling ground truth). Such data allowed us to conduct an extended experimental campaign with two main objectives: (i) assessing the degradation introduced by different sampling policies and rates in aggregated traffic features, irrespectively of the possible applications (e.g. classification, intrusion detection) of such measurements; (ii) evaluating whether flow-level features derived from sampled data are suitable for statistical traffic classification. In the following, we separately summarize the main contributions of this work.

8.1. Impact on traffic measurement and characterization

Our experiments, which considered four sampling-policies (namely systematic, random, stratified and systematic SYN sampling), about 170 traffic features and two distortion metrics, yielded the following findings.

- Unfortunately most of the features are already distorted at low sampling, regardless of the sampling policy.
- Generally a lower degradation affects features based on the inspection of a single packet (such as those related to IP and UDP) with respect to those depending on the analysis of more packets. An exception is represented by those features relying on the inspection of very specific segments (e.g., some TCP options).

DOI: 10.1002/nem

- Regardless of the protocol layer, we isolated a small set of features robust to sampling across all the datasets.
- A sensitivity analysis conducted on this reduced set shows no remarkable advantage of one sampling policy over the others, thus partly contrasting previous studies in favor of random sampling.
- We identify two reasons for the previous finding: 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).
- We spotted 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.

8.2. *Impact on traffic classification*

Regarding traffic classification, we specifically focused on a biased, yet practical, sampling policy (namely systematic SYN sampling) which overcomes the problem pointed out in [14, 15] concerning the statistical significance of the results. Before applying the classification based on C4.5 classification trees, we quantified the information conveyed by features about the application label by means of the information gain metric. The main findings of our experiments follow.

- The cross-investigation of the pure feature distortion and its information-gain loss shows a complex non proportional relation. In particular, there are few features whose information gain remains unchanged under sampling – interestingly they almost coincide with those features derived from a single packet and less distorted by sampling.
- Even more unexpected, some features, though heavily distorted, show an high information-gain score, thus proving important discriminators.
- The information-gain ranking depends on the sampling rate applied, thus suggesting the use of a larger features sets for training the classifier which appears only slightly affected by overfitting.
- Coherently with [15], we show that even in our larger dataset an homogeneous training (i.e., where the same sampling rate has been applied to both training and validation traffic) yields extremely good results, even for harsh sampling (e.g. 1 out of 100 sampling).
- If on the one hand the former observation implies that different training sets should be kept for each sampling rates, on the other hand the classification accuracy degrades gracefully for intermediate heterogeneous solutions (i.e., training on sampled data, but with a different sampling rate). Therefore, good results might be achieved by employing a few training sets obtained with carefully chosen, representative sampling rates.

ACKNOWLEDGEMENT

This work was funded by COST TMA Action IC070. We are grateful to the Editor and the Anonymous reviewer, whose comments helped us improving the quality of the paper.

REFERENCES

1. Amer P, Cassel L. Management of sampled real-time network measurements. *Proc. of IEEE LCN '89*, 1989.
2. Drobisz J, Christensen KJ. Adaptive sampling methods to determine network traffic statistics including the hurst parameter. *Proc. IEEE LCN '08*, Boston, USA, 1998.
3. Claffy KC, Polyzos GC, Braun H. Application of sampling methodologies to network traffic characterization. *Proc. of ACM SIGCOMM '93*, San Francisco, CA, USA, 1993.
4. Duffield N, Lund C, Thorup M. Properties and prediction of flow statistics from sampled packet streams. *Proc. of ACM SIGCOMM IMW '02*, Marseille, France, 2002.
5. Mori T, Uchida M, Kawahara R, Pan J, Goto S. Identifying elephant flows through periodically sampled packets. *Proc. of ACM SIGCOMM IMC '04*, Taormina, Italy, 2004.
6. Kumar A, Xu J. Sketch guided sampling - using on-line estimates of flow size for adaptive data collection. *IEEE INFOCOM '06*, Barcelona, Spain, 2006.

7. Ribeiro B, Towsley D, Ye T, Bolot JC. Fisher information of sampled packets: an application to flow size estimation. *Proc. of ACM SIGCOMM '06*, Rio de Janeriro, Brazil, 2006.
8. Chabchoub Y, Fricker C, Guillemin F, Robert P. Deterministic versus probabilistic packet sampling in the Internet. *Managing Traffic Performance in Converged Networks(LNCS)*, Ottawa, Canada, 07.
9. Hernandez EA, Chidester MC, George AD. Adaptive sampling for network management. *J. Netw. Syst. Manage.* 2001; **9**(4):409–434, doi:http://dx.doi.org/10.1023/A:1012980307500.
10. T Z. Deployment of sampling methods for sla. validation with non-intrusive measurements. *Proc. of PAM '02*, Fort Collins, Colorado, USA, 2002.
11. Mai J, Chuah C, Sridharan A, Ye T, Zang H. Is sampled data sufficient for anomaly detection? *Proc. ACM SIGCOMM IMC '06*, Rio de Janeriro, Brazil, 2006.
12. Brauckhoff D, Tellenbach B, Wagner A, May M, Lakhina A. Impact of packet sampling on anomaly detection metrics. *Proc. of ACM SIGCOMM IMC '06*, Rio de Janeriro, Brazil, 2006.
13. Paredes-Oliva I, Barlet-Ros P, Solé-Pareta J. Portscan detection with sampled netflow. *Traffic Measurement and Analysis (TMA), Springer-Verlag LNCS 5537*, 2009.
14. Jiang H, Moore AW, Ge Z, Jin S, Wang J. Lightweight application classification for network management. *Proc. of ACM SIGCOMM INM '07*, Kyoto, Japan, 2007.
15. Carela-Espaol V, Barlet-Ros P, Cabellos-Aparicio A, Sol-Pareta J. Analysis of the impact of sampling on netflow traffic classification. *Elsevier Computer Networks* 2011; **55**(5):1083 – 1099.
16. Erman J, Mahanti A, Arlitt M, Cohen I, Williamson C. Offline/realtime traffic classification using semi-supervised learning. *Perform. Eval.* 2007; **64**(9-12):1194–1213, doi:http://dx.doi.org/10.1016/j.peva.2007.06.014.
17. Zander S, Nguyen T, Armitage G. Automated traffic classification and application identification using machine learning. 2005; 250 –257, doi:10.1109/LCN.2005.35.
18. Park J, Tyan HR, Kuo CC. Internet traffic classification for scalable qos provision. 2006; 1221 –1224, doi: 10.1109/ICME.2006.262757.
19. Bernaille L, Teixeira R, Akodkenou I, Soule A, Salamatian K. Traffic classification on the fly. *SIGCOMM CCR* 2006; **36**(2):23–26, doi:http://doi.acm.org/10.1145/1129582.1129589.
20. Finamore A, Mellia M, Meo M, Rossi D. Kiss: Stochastic packet inspection classifier for udp traffic. *IEEE/ACM Trans. Netw.* 2010; **18**(5):1505–1515.
21. Tstat, `http://tstat.tlc.polito.it`. URL `http://tstat.tlc.polito.it`.
22. Duffield N. Sampling for passive internet measurement: A review. *Statistical Science* 2004; **19**:472–498.
23. Zseby T, Molina M, Duffield N, Niccolini S, Raspall F. Sampling and Filtering Techniques for IP Packet Selection. RFC 5475 (Proposed Standard) Mar 2009. URL `http://www.ietf.org/rfc/rfc5475.txt`.
24. Paxson V. End-to-end routing behavior in the internet. *SIGCOMM Comput. Commun. Rev.* 1996; **26**(4):25–38, doi:http://doi.acm.org/10.1145/248157.248160.
25. Duffield NG, Grossglauser M. Trajectory sampling for direct traffic observation. *SIGCOMM Comput. Commun. Rev.* 2000; **30**(4):271–282, doi:http://doi.acm.org/10.1145/347057.347555.
26. Choi B, Park J, Zhang Z. Adaptive random sampling for load change detection. *Proc. of ACM SIGMETRICS '02*, Marina Del Rey, CA, US, 2002.
27. Dainotti A, de Donato W, Pescapè A. Tie: A community-oriented traffic classification platform. *TMA*, 2009; 64–74.
28. Aceto G, Dainotti A, de Donato W, Pescapé A. Portload: Taking the best of two worlds in traffic classification. *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, 2010; 1 –5, doi:10.1109/INFCOMW.2010.5466645.
29. Pescapé A, Rossi D, Tammaro D, Valenti S. On the impact of sampling on traffic monitoring and analysis. *International Teletraffic Congress ITC22*, 2010.
30. Group WNR. Auckland-vi traces. `http://www.wand.net.nz/wits/auck/6/auckland_vi.php`.
31. Unibs traces. `http://www.ing.unibs.it/ntw/tools/traces/`.
32. Mantia GL, Rossi D, Finamore A, Mellia M, Meo M. Stochastic Packet Inspection for TCP Traffic. *IEEE International Conference on Communications (ICC'10)*, Cape Town, South Africa, 2010.
33. Gringoli F, Salgarelli L, Dusi M, Cascarano N, Risso F, claffy kc. Gt: picking up the truth from the ground for internet traffic. *SIGCOMM CCR* 2009; **39**(5):12–18.
34. Moore A, Zuev D, Crogan M. Discriminators for use in flow-based classification. *Technical Report*, University of Cambridge, Computer Laboratory, 2005.
35. tcptrace - Official Homepage. URL `http://jarok.cs.ohiou.edu/software/tcptrace/manual.html`.
36. Rossi D, Casetti C, Mellia M. User patience and the web: a hands-on investigation. *IEEE Globecom'03*, San Francisco, CA, USA, 2003.
37. Mellia M, Meo M, Muscariello L, Rossi D. Passive analysis of tcp anomalies. *Elsevier Computer Networks* October 2008; **52**(14).
38. Ribeiro B, Towsley D, Ye T, Bolot JC. Fisher information of sampled packets: an application to flow size estimation. *ACM SIGCOMM IMC'06*, 2006.
39. Pescapé A, Rossi D, Tammaro D, Valenti S. On the impact of sampling on traffic monitoring and analysis. http://www.enst.fr/ drossi/paper/rossi10techrep.pdf 2010.
40. Pescapé A. Entropy-based reduction of traffic data. *Communications Letters, IEEE* Feb 2007; **11**(2):191–193, doi: 10.1109/LCOMM.2007.061068.
41. Mitchell TM. *Machine Learning*. McGraw-Hill: New York, 1997.
42. Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM CCR* 2006; **36**(5):5–16.
43. Weka. URL `http://www.cs.waikato.ac.nz/ml/weka/`.
44. Bermolen P, Mellia M, Meo M, Rossi D, Valenti S. Abacus: Accurate behavioral classification of p2p-tv traffic. *Elsevier Computer Networks* 2011; **55**(6):1394 – 1411.