Identifying key features for P2P traffic classification

Silvio Valenti, Dario Rossi
TELECOM ParisTech
Paris, France
firstname.lastname@enst.fr

***Abstract*—Many researchers have recently dealt with P2P traffic classification, mainly because P2P applications are continuously growing in number as well as in traffic volume. Additionally, in response to the shift of the operational community from packet-level to flow-level monitoring witnessed by the widespread use of NetFlow, a number of behavioral classifiers have been proposed. These techniques, usually having P2P applications as their main target, base the classification on the analysis of the pattern of traffic generated by a host and proved accurate even when using only flow-level data. Yet, all these approaches are very specific and the community lacks a broader view of the actual amount of information of behavioral features derived by flow-level data. The preliminary results presented in this paper try to fill this gap. First of all we define a comprehensive framework by means of which we systematically explore the space of behavioral properties and build a large set of potentially expressive features. Thanks to our general approach, most features already used by existing classifiers are naturally included into this set. Then, by employing tools from information theory and data from packet-level traces captured on real networks, we evaluate the amount of information conveyed by each feature, ranking them according to their usefulness for application identification. Finally we show the classification performance of such a set of features, using a supervised machine learning algorithm.**

I. Introduction

Since their birth in the early Nineties, P2P applications have always represented a concern for network administrators and ISPs, due to the large volume of traffic they are able to generate. Therefore, it is not by chance that the research community invested a lot of effort in designing efficient and accurate methods to identify this traffic [5], [9], [11], [12], [14], [16]–[18], [23], [28], [30], [31]. Things have not changed recently: P2P still powers several killer applications in nowadays Internet (e.g., Spotify, Skype, BitTorrent, and P2P-TV applications) and their classification remains a relevant issue.

Last years, instead, confirmed the trend in flow-level monitoring of operational networks: the growing use of NetFlow, also standardized as IPFIX at the IETF [7], is motivated by a larger scalability with respect to packet-level measurement along with a larger expressiveness compared to the coarse-grained counters of SNMP. Indeed, IPFIX allows to associate a number of *attributes* to flows identified by the classical 5-tuple. Attributes include cumulative packets and bytes counters, flow starting and finishing timestamps, IP type of service, TCP flags, etc. This information, though valuable, differs however from what has typically been exploited to classify *individual flows*. For example, neither the size and direction of the first few packets [5], nor their timestamps [9], nor any fine-grained measures concerning flows [23], nor least of all packet payload [11], [14] is any longer available with IPFIX.

For this reason, with [18], [30]–[32] researchers started developing "behavioral" classifiers. This family of classification techniques, which apply to *flow aggregates* and exploit only coarse-grained information available at transport-level, is very promising; in fact, behavioral classifiers are not only particularly effective with P2P traffic (usually their main target), but they are also able of fine-grained traffic classification [12], [30], [31], while still remaining compatible with the information found in IPFIX records [16], [28].

However, research has so far focused on very specific approaches [12], [16], [28], [30], [31], moreover evaluated on specific datasets, and the community lacks a broad view of the relative importance, in the context of traffic classification, of any feature that can be defined over IPFIX flow-level data. Similarly, as considering a single dataset can bias the evaluation, a careful analysis should explicitly take into account the relative stability of the feature expressiveness over multiple network scenarios.

This work goes in the direction of filling this gap, assessing the reliability of P2P traffic classification based on data from IPFIX-compliant monitors. First of all, we propose a comprehensive framework for the definition of features derivable by IPFIX-records. Our aim is to provide the community with a reference as complete as possible of all behavioral features suited for P2P traffic classification, similarly to what has been done in [22] with flow-level features. We add also the constraint that such features should be fully compliant with IPFIX records. By clearly stating the criteria that guide our definition, we are able to thoroughly explore the space of features and define a long list of potentially expressive characteristics. The resulting framework is general enough to include features of existing classifiers [12], [18], [24], [31], [32], enabling their evaluation as well.

Next, we quantify the amount of information contained in the defined features and exploitable for the classification of P2P traffic. Our analysis is close to what has been done by the authors of [10], which focused on the on the stability of the information carried by traffic flows at the *packet-level*; instead, this work aims at assessing the stability of behavioral features computed at the *flow-level*. We use several packet-level traces from controlled testbeds and real networks, containing traffic from different kinds of P2P application (P2P-TV, VoIP, file-sharing) so that our dataset is representative of various scenarios. From such traffic data, we first compute the features and then evaluate their usefulness for the classification employing metrics from statistics (*Hellinger distance*) and information theory (*mutual information*). After ranking the features in terms of their information content, we briefly presents their classification performance with a supervise machine learning algorithm, namely Support Vector Machines.

II. A Framework for P2P Features Definition

Our classification target is a peer, identified as a socket (or an aggregate of sockets) running on a host. As this peer contacts other peers and exchanges information with them, we suppose that an IPFIX monitor at the edge of the network produces records for all the traffic related to the host. Our framework takes this data as input and derives the features to be used for the classification.

In order to perform a systematic exploration of the feature space, we first introduce a series of criteria, described in the following, to guide the feature definition. We find a good mapping between features used by existing classifiers and our framework, which proves the generality of our approach. While we want to keep our framework as general as possible in its definition, in the experimental part we actually restrict our attention on a smaller subset of the possible features, which is listed in Tab. I and detailed at the end of this section.

A. Timescale

This criterion refers to the temporal duration of the periodical statistics collection, thus dividing time in subsequent timeslots in which features are computed. Observation timescale is subject to the following tradeoff. On the one hand, we would like T to be as small as possible, to support *early classification* for tasks like QoS verification, security and lawful interception. On the other hand, we would like T to be as large as possible for lightweight operation, which would however limit possible applications of the classification to *post-mortem analysis* (e.g accounting, monitoring). Coherently with this requirements, values used in literature range from $T = 5\,s$ in [30] up to $T = 5$ minutes in [31].

Current IPFIX implementations impose further constraints on the choice of T, as they dump statistics on active flows every $T = 30$ minutes, with a configurable minimum of $T = 1$ minute. To have a finer timescale, however, one could use custom implementations on dedicated high-profile device such as Endace [2] or AITIA [1] cards.

B. Entities

We can define the entities involved in a P2P system at different network layers, which in their turn correspond to different levels of traffic aggregation. In fact, a peer can be identified either at L3 by its IP address, at L4 by its port number, or at the endpoint-level by the combination of IP and port.

This is better explained with the help of a simple example. Consider an application running on a host IP_x, receiving all traffic on a single socket on port p_x of L4-protocol type $PT \in \{TCP, UDP\}$. By focusing on different network layers, we can identify the following different entities E

- At the endpoint-level, $E(y) = IP_y : p_y$, by aggregating all IPFIX records $PT : IP_y : p_y : IP_x : p_x$
- At the L3 host-level, $E(y) = IP_y$, by aggregating all IPFIX records $PT : IP_y : * : IP_x : p_x$
- At the L4 port-level, $E(y) = p_y$, by aggregating all IPFIX records $PT : * : p_y : IP_x : p_x$

Basically *endpoint-level* entities correspond to single flows, and have been used in [28], [30], [32]. The other entities, instead, decouple L3 from L4. *Host-level* aggregation, found for instance in [18], [31], may be useful in cases where an application runs multiple sockets (e.g., aggregating several client TCP connections using ephemeral ports, or several UDP sockets with different functions, such as data or signaling). *Port-level* aggregation, instead, might help in evaluating how an application uses the port space (e.g., by using several different random ports, or a single deterministic port). This has been shown to be a good discriminator in [18], [27]. Notice also that, as recently underlined in [19], the port number itself may still be a helpful feature.

C. Granularity and Direction

Since IPFIX records provide counters with different granularities, a trivial criterion regards the level of coarseness of the statics: features can be computed entity-wise E, packet-wise P and byte-wise B.

Another intuitive criterion consists in discriminating *incoming* versus *outgoing* traffic, or aggregating both directions together. Notice that the adopted type of transport layer protocol can cause significant difference in the pattern of traffic observed in the two directions. For example, an application using a connectionless service (i.e. a UDP datagram socket) can easily multiplex all incoming and outgoing traffic over the same endpoint $IP_y : p_y$. Conversely an application employing a connection oriented service (i.e., a TCP stream socket) is likely to receive traffic on a single TCP port (p_y), but it surely spreads the outgoing traffic on different ephemeral ports, whose allocation is controlled by the OS.

D. Categories

The entities involved in the communication with the target peer can be further categorized according to different properties. More on detail, we define some rules C to partition the set of entities $\mathcal{S} = \mathcal{S}_C \cup \overline{\mathcal{S}}_C$. Although in principle the subsets do not need to be disjoint, we believe that requiring $\mathcal{S}_C \cup \overline{\mathcal{S}}_C = \emptyset$ induces more clarity and simplifies the collection of the statistics. We can envisage a number of different properties, related to either the *spatial* or *temporal* domain.

Let us focus on the *spatial* category first. P2P applications offer services built on top of an overlay network which needs to be continuously maintained to handle peers churn. Thus, traffic can roughly be divided in either data or signaling traffic. We consider *contributing* or *data* entities E_d, peers sending or receiving a number of bytes larger than a given threshold. More formally, indicating with B_y the amounts of bytes exchanged with entity $E(y)$, we have $\mathcal{S}_d = \{E(y) : B_y > \beta\}$. Unfortunately, the choice of a proper threshold is not trivial, as it has been shown that good values might be application dependent [27]. In the rest of this work, consistently with [15],

TABLE I
LIST OF P2P TRAFFIC FEATURES USED IN THE EXPERIMENTS, FOR A SINGLE DIRECTION AND A SINGLE TIMESCALE.

	Categories		
Operation	All	New Peers	Data Peers
$O(\cdot)$	$\mathcal{S}$	$\mathcal{S}_n$	$\mathcal{S}_d$
None	E (entities)	E_n	E_d
$O(x) = x$	P (packets)	P_n	P_d
	B (bytes)	B_n	B_d
Difference	$\Delta_t(E)$	$\Delta_t(E_n)$	$\Delta_t(E_d)$
$O(x,t) =$	$\Delta_t(P)$	$\Delta_t(P_n)$	$\Delta_t(P_d)$
$x_{t-1} - x_t$	$\Delta_t(B)$	$\Delta_t(B_n)$	$\Delta_t(B_d)$
Breakdown	-	E_n/E	E_d/E
$O(x_{cat}, x) =$	-	P_n/P	P_d/P
x_{cat}/x	-	B_n/B	B_d/B
Ratio	P/E	P_n/E_n	P_n/E_n
$O(x,y) =$	B/E	B_n/E_n	B_n/E_n
x/y	B/P	B_n/P_n	B_n/P_n
Mean	$\mathrm{E}[\mathbf{P}]$	$\mathrm{E}[\mathbf{P}_n]$	$\mathrm{E}[\mathbf{P}_d]$
$O(\mathbf{x}) =$	$\mathrm{E}[\mathbf{B}]$	$\mathrm{E}[\mathbf{B}_n]$	$\mathrm{E}[\mathbf{B}_d]$
$\mathrm{E}[\mathbf{x}]$	$\mathrm{E}[\mathbf{B}/\mathbf{P}]$	$\mathrm{E}[\mathbf{B}_n/\mathbf{P}_n]$	$\mathrm{E}[\mathbf{B}_d/\mathbf{P}_d]$
Std	$\mathrm{Std}[\mathbf{P}]$	$\mathrm{Std}[\mathbf{P}_n]$	$\mathrm{Std}[\mathbf{P}_d]$
$O(\mathbf{x}) =$	$\mathrm{Std}[\mathbf{B}]$	$\mathrm{Std}[\mathbf{B}_n]$	$\mathrm{Std}[\mathbf{B}_d]$
$\mathrm{Std}[\mathbf{x}]$	$\mathrm{Std}[\mathbf{B}/\mathbf{P}]$	$\mathrm{Std}[\mathbf{B}_n/\mathbf{P}_n]$	$\mathrm{Std}[\mathbf{B}_d/\mathbf{P}_d]$

we use a value $\beta = 12\,KB$, and leave a careful sensitivity analysis as a future work.

We now move to the *temporal* properties. Consider the set $\mathcal{S}_i$ of entities observed at the i-th slot, i.e., $t \in [iT, (i+1)T))$. By comparing $\mathcal{S}_i$ with the previous slot $\mathcal{S}_{i-1}$, we can define the set of *new* entities as $S_n = \mathcal{S}_i \backslash \mathcal{S}_{i-1}$, i.e. the set of peers discovered in the current timeslot. A similar distinction can be found in many works on P2P traffic analysis [6], [27], or P2P traffic classification [31].

While in this work we just consider the above two rules, it is worth mentioning a few other partitions related to the temporal domain. For instance we can define the set of k-persistent neighbors as the set of peers that have been seen in (at least) k consecutive rounds $\mathcal{S}_{k-per} = \cap_{j=i}^{i-k} \mathcal{S}_j$. Symmetrically we could define the set of k-recurring peers, i.e., the peers that are to be found in the current and in the k-th previous slot, as $\mathcal{S}_{k-rec} = \mathcal{S}_i \cap \mathcal{S}_{i-k}$. Besides, many more complex presence indicator such as those defined in [26] could be evaluated.

E. Operations

Finally, a wide range of computation can be performed on the gathered counters data, ranging from very simple to rather complex operations. As examples of the latter, in [30], a probability mass function is built starting from the counts of packets and bytes exchanged by the target peer with the other entities. In [12], instead, the Autocorrelation function (ACT) and the discrete Fourier transform (DFT) is applied to the time series of entity counts, of data rates exchanged with a given entities, and of start and end time of flows.

Within the scope of this work, however, we limit the analysis to the following few simple operations $f(\cdot) : \mathbb{N}^m \to \mathbb{R}$, that produce a single scalar value.

- *None*, use the raw count as feature.
- *Temporal difference* with respect to the previous slot (e.g., the rate at which the number of packets received is changing $\Delta_t(P) = P_t - P_{t-1}$)
- *Category breakdown* per-category breakdown (e.g., the percentage of *new* entities E_{new}/E).
- *Ratio* of different counters for a given entity (e.g., B_y/P_y the mean packets size of a given entity y).
- *Spatial mean and standard deviation* of a counter over a set of entities (e.g., mean number of packets per peer $\mathrm{E}[\mathbf{P}]$)

Despite their simplicity, quite a few of these operations have already been successfully employed for traffic classification, for example in [17], [18], [24]. We point out that this list is clearly not exhaustive (e.g., temporal means and other statistics can naturally be defined). However, we believe that the merit of the framework is not weakened by considering, for the time being, a small but well-defined list, and leaving the thorough exploration of this criterion as future work.

Let us clearly state here the set of features we are going to analyze in the experimental part. First of all, we decided to consider two timescales $T \in \{5, 120\}\, s$, outgoing and incoming traffic separately, and endpoint-level entities. Finally, operations and categories are summarized in Tab. I. Features are organized in columns according to the category they pertain to (i.e. all, new and data entities), and in rows according to operation performed to calculate them. Notice that we use counters of all possible levels of granularity (entity, packets, bytes), along with all their meaningful combinations in computing ratios and statistical indexes. Overall the final set is composed of 102 features.

III. THE FRAMEWORK IN ACTION

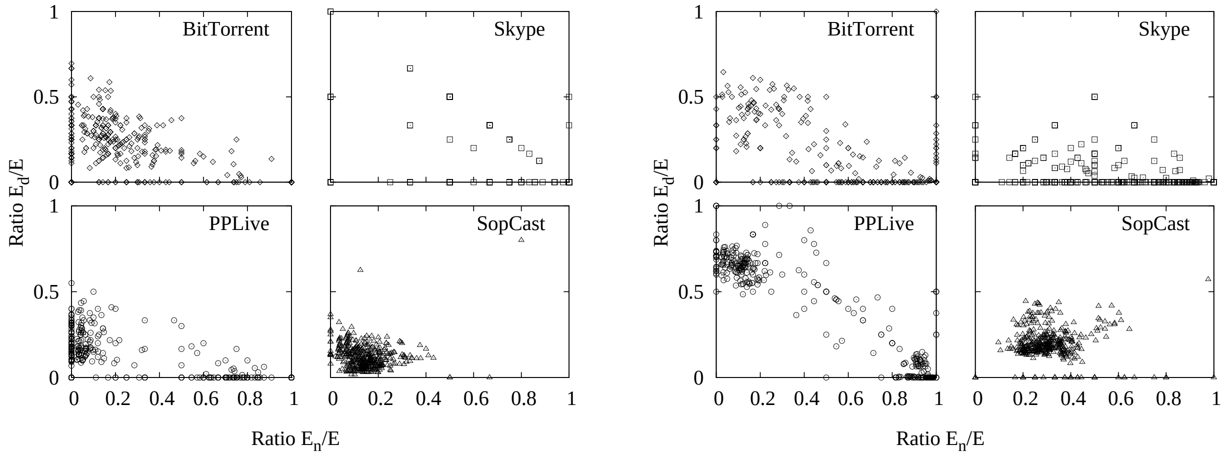
A. Dataset

In order to obtain robust results, we resorted to a large dataset of packet-level traces with associated ground-truth, composed by three distinct parts. For lack of space we omit the details, for which we rather refer the reader to [4], [11], [28]. Here, we only briefly summarize its characteristics: it consists of 7 applications (see below for further information), for a total of $900K$ distinct IP addresses, $120M$ packets and $50G$ bytes. Traces have been collected with a mixture of active (from 7 vantage points in Europe) and passive methodologies (from three different networks), and are thus a good starting point for the evaluation[1]. The ground-truth for the passive traces is provided either by a DPI classifier and manual inspection, or by GT [13].

Let us spend some words on the target applications. In order to gather consistent results, we tried to include in our dataset a representative sample of the whole spectrum of P2P applications. More on details, the dataset is made up of traffic from four *P2P-TV* applications (PPLive, TVAnts, SopCast, Joost [2]), two *file-sharing* applications (Edonkey, BitTorrent) and a *VoIP* application (Skype).

[1] Active testbed and part of passive traces are available upon request, respectively at [3] and [4]. The other passive traces are instead protected by NDA

[2] Traces date back May 2008, when Joost was still exploiting P2P for video distribution



(a) $T = 5\,s, HD = (0.5, 0.11), I = (1.15, 0.06)$

(b) $T = 120\,s, HD = (0.52, 0.28), I = (1.17, 0.39)$

Fig. 1. Scatter plots of the breakdown of new entities versus the breakdown of data entities for four applications and two timescales. Captions report the values of the statistical metrics for the feature on x and y axis.

B. Methodology

In this section we describe the procedure followed to evaluate the importance of each feature for traffic classification. We started by extracting IPFIX-records from the packet traces. Then we aggregated the records related to the relevant endpoints and compute the features. The result is a list of couples $(Y, \mathbf{X})$, where Y is the application label, and $\mathbf{X}$ is the vector of features listed in Tab. I, computed on both incoming and outgoing traffic. Afterwards, similarly to [10], we extracted a random subset of all data, in such a way that it finally contains the same number of samples for each application–corresponding to about 6 hours worth of traffic for each application. Besides facilitating our analysis, this process removes any bias deriving from unbalanced traffic mixture.

Subsequently, we evaluated the expressiveness of each feature by means of two statistical metrics. The first metric is the *mutual information* $I(X, Y)$ [8] between a feature X and the application label Y. $I(X, Y)$ measures the amount of information shared by the two random variables X and Y, or, in our case, how much the knowledge of feature X tells about the protocol label Y. This quantity is always positive and usually measured in bit. If the two variables X and Y are completely unrelated, their mutual information is zero. Conversely, a feature X is a perfect discriminator if the condition $I(X, Y) = H(Y)$ is met, where $H(Y)$ is the entropy of the protocol label (i.e. the number of bits needed to perfectly describe Y). In our case of 7 applications represented by the same number of samples we have $H(Y) = \log_2(N) \sim 2.8$.

The second metrics, the *Hellinger distance* $HD(p, q)$, measures the similarity between the two probability distributions p and q. It takes values in $[0, 1]$: a score of 0 means that the two probabilities are identical, while a score of 1 means that they are completely different. We already used this metric to asses the impact of sampling on flow-level measurements in [25] and to assist traffic classification in [30]. In this context we actually evaluate $\mathrm{E}_Y[HD(p_{X|Y}, p_X)]$, i.e. the expectation over all classes Y of the distance between the marginal distribution p_X of feature X, and the conditioned distribution $p_{X|Y}$. Intuitively, if all the conditioned distributions $p_{X|Y}$ related to the same feature are distant from the marginal distribution $p(x)$, then they can be easily distinguished. Given this consideration, the larger the score of this metric, the more likely it is that the feature is a good discriminator. Notice also that $I(X|Y)$ can be expressed in a similar way as $I(X, Y) = \mathrm{E}_Y[D_{KL}(p_{X|Y}, p_X)]$, where D_{KL} denotes the Kullback-Leibler divergence.

To calculate the above metrics we need an estimation of the distributions p_X and $p_{X|Y}$. All along this work, we use empirically probability mass functions: in other words we approximate the real distribution of a feature with its histogram, using a linear binning over the support of the feature itself. This is known to introduce some bias, especially in the estimation of the mutual information; therefore, to mitigate this effect, we apply the Miller-Meadow correction [21] to our results. We do not apply any correction to the Hellinger distance, which, in our experience [25], behaves well with empirical distributions.

C. Example

Before applying the above metrics to the whole set of features, we show a preliminary example of their discriminating power. In Fig. 1 we report the scatter plots of two features, namely the breakdown of new entities (E_n/E) on the x-axis, and the breakdown of data entities (E_d/E) on the y-axis, considering four applications and both timescales $T \in 5, 120\,s$. Although for clarity sake we do not represent all the 7 applications, nevertheless pictures include at least one example for each service (P2P-TV, filesharing, VoIP).

At first glance, the impression is that each application generates a distinct pattern, as points cluster in different regions of the plane. The larger time scales seems to yield better results as point clouds appear to be very well differentiated, with few regions of intersection between each other. Notably, a bimodal behaviors of PPLive is highlighted, with two distinct clouds of points corresponding exactly to the two typical activities of a P2P application: discovering new peers (i.e. low percentage of data peers, but high percentage of new peers), and transferring data (i.e. high percentage of data peers, but low percentage of new peers). Notice, instead, that for $T = 5\,s$ the region of space near the origin, i.e. low percentage of both new and data peers, is more or less common to all applications.

A general observation, which is confirmed by the values of the statistical metrics reported in the figure captions, is that applications are better separated along the x-axis than along the y-axis. In fact both metrics present larger scores for the breakdown of new peers, than for the breakdown of data peers. It can be argued that considering the information contained in the *combination* of the two features would provide further insight in this kind of analysis. Yet, in this paper, we prefer to adopt a simpler approach and evaluate each feature on its own, leaving the investigation of this issue as a future work.

D. Experimental Results

We now extend our analysis to the whole set of features. Fig. 2 reports the scores assumed by the two metrics for each feature, computed with a timescale of $T = 5\,s$. The top row of graphs reports the mutual information, while the bottom row shows the mean Hellinger distance. Each row is composed by three plots, related to the different granularities of the counters (from left to right: entities, packets and bytes); each plots is further divided in three histograms, one for each category of entities (all, new, data). Finally, each possible operation is represented by a bar, with white bars denoting features calculated over the incoming direction, while gray bars report values for the outgoing direction. Because of space constraints, we omit results for $T = 120\,s$, as they show a qualitative similar behavior.

Like in the preliminary example, the two statistical metrics yield coherent results: they assume high values for the same features. Most of information seems to be captured by the entity-wise and byte-wise features, while packet-wise features appear less relevant, except for the breakdown operation, which interestingly yields the most valuable features. We also observe that the data category has, on average, better scores than the new category, revealing itself a good discriminator, despite its being threshold based.

Nevertheless, the most important conclusion that can be drawn is that no feature is a significant discriminator on its own. Notice, in fact, that only a few features exceed the score of one bit for the mutual information, or the score 0.5 for the Hellinger distance. This was somehow expected since the features considered are very simple. To compensate this issue, a behavioral classifier must either choose a wise combination of features (as we do in the next section), or use more complex operations on the counters (e.g. what found in [12], [30]), which try to capture more specific properties of the pattern of traffic. In our future work we intend to apply our methodology to this second kind of features as well.

E. Classification

In this section we present the performance of our features when they are used for traffic classification. We resort to Support Vector Machine as classification algorithm, as it was proved particularly accurate for network traffic classification [19].

TABLE II
(A) TOP FEATURE INDIVIDUATED BY EACH METRIC AND (B) CLASSIFICATION ACCURACY(%) FOR $T = 5\,s$

(a)

Feature set	
HD	I
E_n/E	B_d/P_d
P_n/P	E[$\mathbf{B}_d/\mathbf{P}_d$]
P_d/P	P
B/P	E_d/E
B_c/P_c	-
E[$\mathbf{B}/\mathbf{P}$]	-
E[$\mathbf{B_d}/\mathbf{P_c}$]	-
$\star P_n/P$	-
$\star B_n/B$	-

(b)

	Accuracy	
App.	HD	I
PPLive	29.2	7.6
TVAnts	40.5	30.95
SopCast	94.1	83.9
Joost	97.1	98.6
Edonkey	65.4	53.3
BitTorrent	94.7	93.4
Skype	70.6	57.7
Overall	70.2	60.8

When dealing with such a large number of features, it is a common practice to use feature selection algorithms, which remove useless or possibly misleading features. Yet, in this work we decided to use the results from the previous analysis and implement a naïve feature selection, using a simple greedy algorithm. We first perform the classification using the most relevant feature according to one statistical metric. Then we repeat the classification including the second most relevant feature, and we proceed incrementally adding new features in descending order of score of the metric, until no further improvement of the classification performance is appreciated.

In Tab. II-(a) we list the features identified by our algorithm for the two statistical metrics, where starred features refers to the outgoing direction. Tab. II-(b) shows, instead, the per-application classification accuracy for $T = 5\,$s. Both metrics select a small number of features, achieving however a good accuracy, given also the simplistic approach adopted. Interestingly among selected features there is a majority of ratios and breakdowns, which means that relative values are more expressive than absolute ones. Greedy feature selection appears to work well with HD, which exploits 9 features and achieves a better accuracy, whereas I seems to need more sophisticated selection algorithms. Among the applications, PPLive is the most difficult to recognize, while extremely good performance is shown for Joost, Edonkey and BitTorrent.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we presented two main results. First we introduced a coherent and comprehensive framework for the definition of behavioral features for the classification of P2P traffic. In its definition, we assumed that only flow-level measurements are available. By clearly stating the criteria at the base of feature definition, which take into account both the nature of input data (IPFIX flow counters), and of the target traffic (meshed P2P systems), we obtained an extremely general framework, which, in our opinion, could represent a valuable reference for the research community.

In the second part of this work, we performed an analysis of the amount of information carried by the defined features, using two statistical metrics. We considered a large set of about 100 features, with two different timescales for statistic

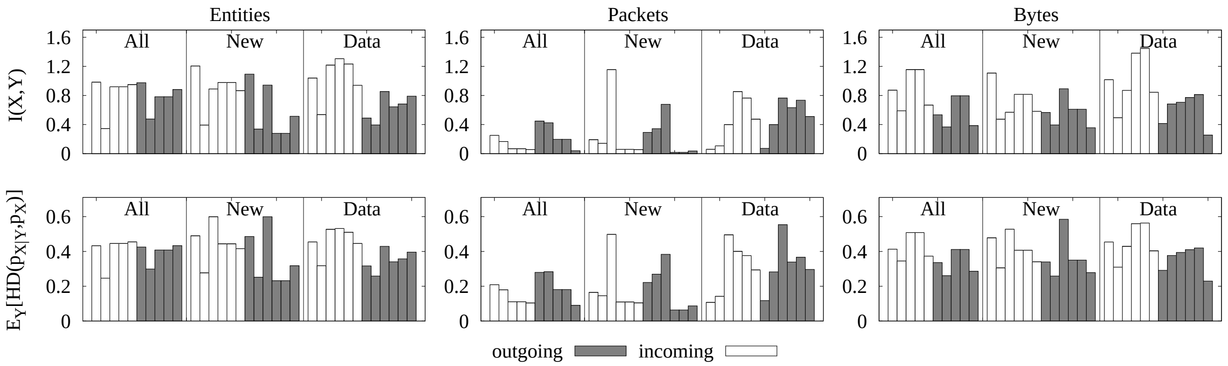


Fig. 2. Values of $I(X,Y)$ and mean HD for $T = 5\,s$ for incoming and outgoing direction of traffic. Bars in each histogram represent the different operations in the order listed in Tab. I.

collections, over a large dataset of traces generated with active and passive methodologies and including 7 P2P applications. Results showed that each feature in isolation conveys only a limited amount of information, but that carefully chosen combinations of features might represent good discriminators for traffic classification. In fact, by means of a naïve feature selection scheme based on a greedy algorithm, we are able to achieve 70% of accuracy, by using only the best 9 features according to the statistical metrics.

Our future work will be addressing first of all the extension of the set of features, taking into account different levels of aggregation (e.g. host-level and port-level) along with more complex operations on the counters, as these have been shown to enhance classification accuracy [12], [30]. We also want to improve the estimation of mutual information, for instance by employing non parametric methodologies [29], possibly in combination with specific feature selection schemes based on this estimation, like e.g. [20], which will enable a better assessment of the classification performance.

Acknowledgements

This work has been funded by the Celtic project TRANS.

References

[1] Aitia. http://www.aitia.hu.

[2] Endace. http://www.endace.com.

[3] European fp7 napawine project. www.napawine.eu.

[4] Univ. Brescia traces. http://www.ing.unibs.it/ntw/tools/traces/.

[5] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In *ACM CoNEXT '06*, Lisboa, Portugal, Dec 2006.

[6] D. Ciullo, M. A. Garcia, A. Horvath, E. Leonardi, M. Mellia, D. Rossi, M. Telek, and P. Veglia. Network awareness of p2p live streaming applications: a measurement study. *IEEE Trans. on Multimedia*, 12(1):54–63, 2010.

[7] B. Claise. Specification of the IP Flow Information Export Protocol for the Exchange of IP Traffic Flow Information. RFC 5101, Jan 2008.

[8] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley, New York, 1991.

[9] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Comput. Commun. Rev.*, 37(1):5–16, 2007.

[10] A. Este, F. Gringoli, and L. Salgarelli. On the stability of the information carried by traffic flow features at the packet level. *ACM SIGCOMM Comput. Commun. Rev.*, 39(3):13–18, 2009.

[11] A. Finamore, M. Mellia, M. Meo, and D. Rossi. Kiss: Stochastic packet inspection classifier for udp traffic. *IEEE Trans. on Net.*, to appear 2010.

[12] T. Z. J. Fu, Y. Hu, X. Shi, D.-M. Chiu, and J. C. S. Lui. Pbs: Periodic behavioral spectrum of p2p applications. In *Proc. of PAM '09*, Seoul, South Korea, Apr 2009.

[13] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and k. c. claffy. Gt: picking up the truth from the ground for internet traffic. *ACM SIGCOMM Comput. Commun. Rev.*, 39(5):12–18, 2009.

[14] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. Acas: automated construction of application signatures. In *ACM MineNet '05*, Philadelphia, Pennsylvania, USA, Aug 2005.

[15] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. A measurement study of a large-scale p2p iptv system. *IEEE Trans. on Multimedia*, 9(8):1672–1687, 2007.

[16] H. Jiang, A. W. Moore, Z. Ge, S. Jin, and J. Wang. Lightweight application classification for network management. In *ACM SIGCOMM Internet network management (INM '07)*, Kyoto, Japan, Aug 2007.

[17] T. Karagiannis, A. Broido, M. Faloutsos, and K. claffy. Transport layer identification of p2p traffic. In *ACM IMC '04*, pages 121–134, Taormina, Sicily, Italy, Oct 2004.

[18] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM Comput. Commun. Rev.*, 35(4), 2005.

[19] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *Proc. of ACM CoNEXT 2008*, Madrid, Spain, 2008.

[20] N. Kwak and C.-H. Choi. Input feature selection by mutual information based on parzen window. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(12):1667–1671, 2002.

[21] G. A. Miller. Note on the bias on information estimates. In *Information Theory in Psychology, Problems and Methods.*, pages 95–100.

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

[23] A. W. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. In *ACM SIGMETRICS '05*, Canada, Jun 2005.

[24] M. Perényi, T. D. Dang, A. Gefferth, and S. Molnár. Identification and analysis of peer-to-peer traffic. *Journal of Communications*, 1(7):36–46, 2006.

[25] A. Pescapé, D. Rossi, D. Tammaro, and S. Valenti. On the impact of sampling on traffic monitoring and analysis. In *Proc. of ITC22*, Amsterdam, The Netherlands, Sep 2010.

[26] D. Rossi, M. Mellia, and M. Meo. Understanding Skype signaling. *Elsevier Computer Networks*, 53(2):130–140, 2009.

[27] D. Rossi and E. Sottile. Sherlock: A framework for P2P traffic analysis. In *IEEE P2P'09*, Seattle, WA, USA, Sep 2009.

[28] D. Rossi and S. Valenti. Fine-grained traffic classification with Netflow data. In *TRaffic Analysis and Classification (TRAC) Workshop at IWCMC '10*, Caen, France, Jun 2010.

[29] S. Shwartz, M. Zibulevsky, and Y. Y. Schechner. Fast kernel entropy estimation and optimization. *Elsevier Signal Process.*, 85(5):1045–1058, 2005.

[30] S. Valenti, D. Rossi, M. Meo, M. Mellia, and P. Bermolen. Accurate, Fine-Grained Classification of P2P-TV Applications by Simply Counting Packets. In *Traffic Measurement and Analysis (TMA), Springer-Verlag LNCS 5537*, pages 84–92, May 2009.

[31] C.-C. Wu, K.-T. Chen, Y.-C. Chang, and C.-L. Lei. Peer-to-peer application recognition based on signaling activity. In *Proc. of IEEE ICC '09*, Dresden, Germany, May 2009.

[32] K. Xu, Z.-L. Zhang, and S. Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. *ACM SIGCOMM Comput. Commun. Rev.*, 35(4):169–180, 2005.