On the Stability of the Information Carried by Traffic Flow Features at the Packet Level

Alice Este, Francesco Gringoli, Luca Salgarelli*
DEA, Università degli Studi di Brescia, Italy
Email: <first.last>@ing.unibs.it

ABSTRACT

This paper presents a statistical analysis of the amount of information that the features of traffic flows observed at the packet-level carry, with respect to the protocol that generated them. We show that the amount of information of the majority of such features remain constant irrespective of the point of observation (Internet core vs. Internet edge) and to the capture time (year 2000/01 vs. year 2008). We also describe a comparative analysis of how four statistical classifiers fare using the features we studied.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations

General Terms

Measurement, Experimentation

Keywords

Traffic classification, transport layer

1. INTRODUCTION

Traffic classification, the branch of traffic measurement that studies mechanisms to associate traffic flows to the applications that generated them, in the last few years has focused on the statistical analysis of measurable features, such as packet size or flow duration [1, 2, 3]. Many classification systems have been proposed, and their effectiveness has been proven on a series of different traffic traces, collected in various network locations and in various time periods.

In this paper we examine the capability of each traffic flow feature, measured at the packet level, of discriminating the application that has generated the flow, giving an answer to the following basic questions: does this capability change over time? Is it affected by the observation point?

In order to answer those questions, we measure, using statistical parameters, the amount of information carried by the features in three traffic traces, one collected in 2008 at the access link of our Faculty's network, one in 2001 at the gateway of another large university and one collected in 2000 at an Internet Exchange interconnecting large Service Providers.

*This work was supported in part by a grant from the Italian Ministry for University and Research (MIUR), under the PRIN project *RECIPE*.

The results of the analysis show that the information carried by the main packet-level features of Internet traffic flows tends to remain rather constant both in space and in time. A few exceptions also emerge from our analysis, and are discussed in the rest of this work. Finally, we present a brief comparative analysis of how several classification mechanisms fare in relation to the information carried by the features evaluated earlier.

The paper is organized as follows. The remaining part of this Section briefly describes related works. In Section 2 we introduce the methodology we used to measure the amount of information associated to traffic flow features. Section 3 describes the data sets over which we run the experiments. In Section 4 we analyze the stability in time and in space of the information carried by the features. In Section 5 we show the accuracy results of four classification systems, applied to the analyzed quantities. Section 6 concludes the paper.

1.1 Related work

Some recent papers [4, 5, 6] have used statistical measurements in the field of traffic classification. However, the use of such measurements in those works is specifically devoted to feature selection, i.e., to determining which features work best for classification purposes. The objective of our work is different: we want to analyze whether the contribution of each feature in discriminating the protocol classes is the same in different network locations and if it does not change in time. For obvious space constraints we omit references to the many papers that in recent years have focused on statistical traffic classification: besides the fact that, to the best of our knowledge, none of them had the explicit objective of analyzing the stability of the information content of traffic features, they would be too many to mention in this context. Finally, while [7] presents a feature selection mechanism based on the perceived stability of the information carried by each feature, it does not analyze comparatively the amount of information carried by them.

2. METHODOLOGY

2.1 Definitions: flows, features and classes

We define as *flow* the bi-directional ordered sequence of packets two end-points exchange, identified by host IP addresses and TCP ports. In this paper we focus on TCP traffic, that represent more than 95% of the bytes in the three traffic traces we used in this paper. We will extend this work to UDP in a future paper.

Each flow begins with the TCP three-way-handshake, and we label as *client* the end-point that sends the first packet with the SYN bit flag set to 1. We exclude from the analysis connections that do not correctly complete the three-way-handshake. We also exclude the flows for which we do not see the complete handshake through the monitoring node: in practice, this means considering almost exclusively traffic that is routed symmetrically. Once again, we leave the analysis of asymmetric traffic to a future work. We do not apply operations of de-fragmentation or re-ordering of the packets: the sequence is in the same order the monitoring network node collects it.

We isolate the packets corresponding to distinct flows, and from the first three packets, representing the three-way-handshake phase, we extract the following *features*:

- $RTT = (rtt_s, rtt_c)$, the RTT estimation between the server (client) and the monitoring node. The rtt_s is the time interval between the receiving of the SYN packet (at the monitoring device) and the server response (SYN-ACK), while rtt_c is the interval between the forwarding of the SYN-ACK packet and the client response (ACK). The three-way-handshake packets, sent in the early stage of each connection, are mainly affected by the distance of the end hosts from the monitoring device and by the network traffic conditions. They usually do not carry payload bytes, therefore they allow us to achieve a good estimation of the response time of the two end-points. Note that including packets following the three-way-handshake in this estimation would corrupt it for our purposes, since their RTT could be more affected by "noisy" factors such as congestion control, packet loss, etc.
- $O = (o)$: when it is possible, it denotes the position of the client with respect to the monitoring node, indicating if the client is inside or outside the range of IP addresses of the LAN. Therefore, this is a binary feature.

We extract an additional set of features from the first N packets of each flow, considering only packets that carry a non-zero payload. In fact, empty packets are mostly used to transmit connection state information, e.g., to acknowledge received data or to keep alive a session. Therefore they are due to the transport layer internals, as opposed to being linked to the way each specific application operates. For example, the inter-arrival time between a data segment and an empty ACK packet carries more information about the status of the TCP state machine rather than the way an application-layer protocol operates. On the contrary, we expect that the inter-arrival times between consecutive data packets can differentiate a bulk transfer application from an interactive one. More on this in Section 4.

We will use i to indicate the index of a packet in the flow sequence, as seen by the capturing node, and excluding packets that do not carry TCP payload bytes. The additional set of features is composed of:

- $S = (s_1, s_2, ...s_N)$, where s_i is the payload size of the i-th packet, taking value in the range [1; 1460].
- $T = (t_1, t_2, ...t_N)$, where t_i is the inter-arrival time between the $(i-1)$-th and the i-th packet, with t_1 always void.
- $D = (d_1, d_2, ...d_N)$, where d_i is the end-point transmitting the i-th packet (the client or the server).

Finally, with the term *class* we denote a set of flows that were generated by the same application protocol. We obtain the *ground truth* of the protocol that is responsible of the generation of a flow using payload-based pattern matching techniques [8] and TCP port numbers, as we will explain in Section 3. Therefore, to each flow we also associate a feature Y, denoting the protocol that generated the corresponding packet sequence.

2.2 Information measures

In this paper we apply statistical measurements to evaluate the amount of information carried by the flow features.

We use *mutual information* measure $I(X;Y)$ [9] between the extracted features X and the protocol index Y, that estimates the information contribution that each feature provides regarding the protocol. $I(X;Y)$ is a measure of the dependence between the values of X and Y, it is a non-negative quantity and it is zero only when X and Y are independent. We also measure the *conditional mutual information* $I(X;Y|Z)$ between X and Y, given the knowledge of another feature Z, that represents the reduction of uncertainty of Y due to the knowledge of X, when the value of Z is given. It is non-negative and it is zero only when X and Y are independent conditioned on knowledge of Z.

To evaluate these measures we need to know an estimation of the densities of the features. We approximate these functions adopting multivariate non-parametric kernel density estimations [10], using a Gaussian kernel function. Given a class, we approximate the feature distributions starting from a finite set of n samples $T_1 = (\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n)$, corresponding to n flows belonging to the class. We obtain the density estimations, using the Expectation Maximization (EM) algorithm to estimate, with an iterative procedure, as proposed in [11], maximizing the likelihood function [10] on a different set of n observations T_2 of the same class.

Using non-parametric kernel density estimations we build effective models using just around one thousand flows for each class of the data sets described below, according to the numerousness of samples we extracted from the traces. We experimentally validate the accuracy of the models by ensuring that the measures we obtain using them remain constant even increasing the number of flows beyond the one thousand mark.

3. DATA SETS

In this paper we use three traffic traces collected in three different network positions: the first is the gateway interconnecting our Faculty's network with the Internet (UNIBS data set), the second is the border router of the University of Auckland (AUCKLAND data set) and the third is an Internet Exchange Point serving as a peering point to large Internet Service Providers (NZIX data set). We report in Table 1 the main characteristics of the traces.

The choice of these three data sets has been made carefully, and is instrumental in the kind of analysis we want to develop in this paper, considering mainly three aspects. First, they have been captured in different locations, heterogeneous in terms of link capacity and service offered (Internet core vs. Internet edge). Second, they have been collected several years apart. Third, for different reasons, as

Dataset	Date	Type	Dur.	Volume
UNIBS (T_1)	2/Apr/08	LAN recent	5 h	36.6 GB
UNIBS (T_2)	4/Apr/08	LAN recent	6 h	35.7 GB
AUCKL. (T_1)	8-10/Jun/01	LAN old	65 h	9.7 GB
AUCKL. (T_2)	11-13/Jun/01	LAN old	45 h	9.4 GB
NZIX (T_1)	5-6/Jul/00	backbone old	32.5 h	20.4 GB
NZIX (T_2)	7-8/Jul/00	backbone old	48 h	24.0 GB

Table 1: Description of the traces. Note that the UNIBS traces include the first 250 bytes of each frame, while all the other traces are "header only".

explained below, the ground truth wrt. the application protocols can be ascertained with a relatively high degree of confidence for these traces. We now describe the data sets in more details.

3.1 UNIBS data set

The packet traces composing the UNIBS data set were collected at the border router of our Faculty's network. Our network infrastructure comprises several 1000Base-TX segments routed through a Linux-based dual-processor box and includes about a thousand workstations with different operating systems. The traffic traces were captured on the 100Mb/s link connecting the edge router to the Internet, for a total of 72.3 GB collected by running Tcpdump [12], as we show in Table 1. This data set includes the traffic of two days captured in the middle hours, when the link load is higher. All the traffic exchanged between the LAN and the Internet necessarily passes through the collecting device. We use the trace collected on the first day to build the T_1 set (see Section 2.2), while the second day provides the T_2 set.

Since we have full monitor access to this router, we stored the first 250 bytes of every frame. We applied pattern-matching mechanisms on the payloads to assess the actual application that has generated each TCP flow, in some cases with the addition of manual inspection. Because we have the first payload bytes, we consider the ground truth, i.e., the value of the feature Y, relatively reliable for this data set for plaintext protocols. We also consider flows matching the SSL pattern, in this case identifying them by the TCP port number. Therefore, we include in the data set flows to port 443 as HTTPS and to port 995 as POP3S.[1]

We show in Table 2 the protocol classes we selected; they belong to different application types: web browsing, mail services, P2P and interactive. They were chosen because they are responsible for the generation of the most part of the traffic, and because they span from interactive to bulk-transfer protocols. In addition, most of them are easily identifiable with pattern-matching methods with a satisfactory degree of accuracy.

Since we need a few packets with non-zero payload for each flow to carry out meaningful experiments, we choose to consider flows with at least six packets, which, for this data set, means including in the analysis more than 99% of the captured traffic.

Although the capturing mechanism for this data set (Tcpdump) is not as precise as one would like, we use these traces in this paper because (i) they are amenable to pattern matching for deriving ground truth information, (ii) they are quite recent, as opposed to the other two data sets, and (iii) they are representative of the LAN scenario (edge vs. core).

[1]We noted that this data set does not include much "hidden" traffic, i.e., TCP traffic that does not match any of the patterns we selected: this amounts to only 0.3% of bytes (around 7% of flows).

3.2 AUCKLAND data set

The second set we consider, named AUCKLAND, is a 5-day trace collected at the University of Auckland in June 2001, available at [13, 14]. The traces include only the header bytes, with a maximum amount of 64 bytes for each frame, while the application payload is fully removed. The header traces were captured with a GPS synchronized mechanism using a DAG3.2E card connected to a 100Mbps Ethernet hub interconnecting the University's firewall to their border router.

Due to the anonymization of the traces, in this case we cannot apply pattern-matching procedures to derive ground truth information. The only information available is the TCP port number, which is preserved by the anonymizer. Each flow is thus assigned to the class identified by the server port, with the exception of the FTP-DATA class (in active mode), where the standard client port (20) is used. In Table 2 we report the list of ports with the larger number of flows. For the AUCKLAND data set we consider the port information a relatively reliable source of ground truth, because we estimate that in June 2001 the circulation in the network of P2P protocols, that is one of the main causes of the loss of reliability of the TCP port in identifying the protocol, was very limited. In the trace, in fact, we observe the absence of ports usually used by P2P protocols, except for a negligible amount of flows on port 6699, used by the WINMX file-sharing protocol and on port 6346 used by the GNUTELLA protocol.

3.3 NZIX data set

The third data set is a four day traffic trace obtained monitoring the New Zealand Internet Exchange (NZIX) in July 2000 [15, 14]. These traces also include only the header bytes, while the application payload is fully removed. The header traces were captured using a DAG3.2E with a GPS synchronized mechanism at the New Zealand Internet Exchange, which served as a peering point for large New Zealand ISPs. At the time, NZIX consisted of two Cisco 2926 26 port 10/100 auto-sensing Ethernet switches running spanning tree for redundancy. These trace are publicly available in anonymized form, therefore also in this case we use the TCP ports to derive ground truth information.

Analyzing only the traffic flows that correctly complete the three-way-handshake, we exclude all the flows that have traffic in only one direction through the monitoring device. In the UNIBS and AUCKLAND data sets this cannot happen, because all the traffic of the LAN toward the Internet passes through the capture node, but the core of the network can be subjected to asymmetric routing. This selection procedure allows us to exclude the majority of these cases, restricting the analysis almost exclusively to flows captured in both their transmission directions[2]. The choice of evaluating only bi-directional flows allows us to compare the features extracted from the three traces in a homogeneous way. We plan to extend our analysis to asymmetrically-routed flows in future works.

[2]In the NZIX trace bi-directional flows account for 49.6% of the captured traffic in bytes (25.7% of flows).

protocol	2 Apr '08	4 Apr '08
http	195268	224904
ssl (443)	28041	26813
edonkey	22457	24702
smtp	12060	11816
pop3	4324	4745
ssl (995)	2326	2932
bitTorrent	2375	1752
msn	1310	1607

port	protocol	8-10 Jun '01	11-13 Jun '01
80	http	870625	994872
443	https	93378	134835
25	smtp	66607	92278
110	pop3	14611	20599
22	ssh	19131	13651
21	ftp-ctrl	5224	5172
20	ftp-data	3244	3854
143	imap	2524	2115

port	protocol	5-6 Jul '00	7-8 Jul '00
80	http	502777	554955
443	https	108202	117157
110	pop3	79358	101285
25	smtp	78270	76011
21	ftp-ctrl	1540	1561
20	ftp-data	2751	2933
27	nsw-fe	1363	2047
119	nntp	1033	959

Table 2: Number of flows with at least 6 packets with payload in the UNIBS data set (left), the AUCKLAND data set (center), the NZIX data set (right).

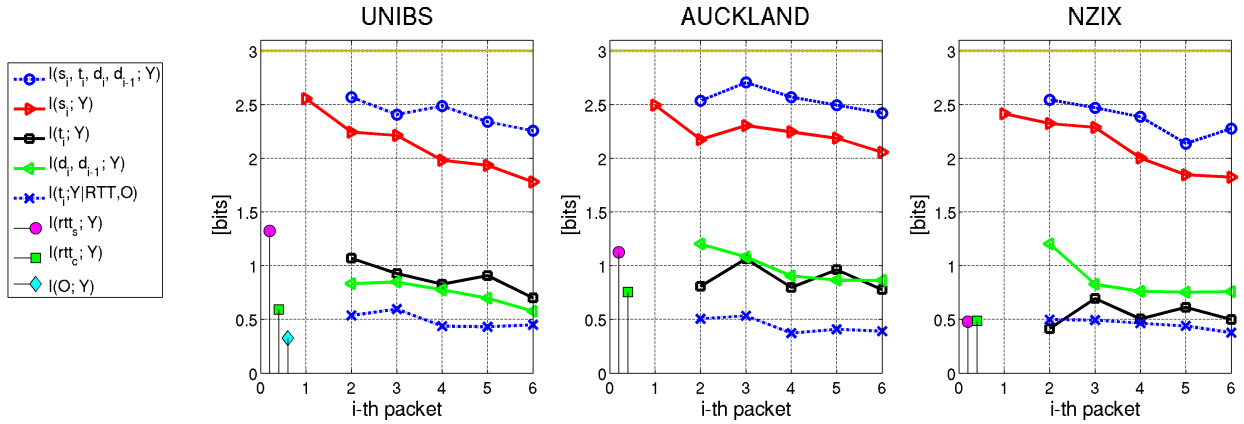

Figure 1: Mutual information between the features extracted considering the first packets of each flow and the application generating the flow (measured in bits).

4. FEATURE ANALYSIS

In this Section we investigate the information contribution each packet-level feature brings in determining the application protocol behind each traffic flow.

In Figure 1 we show, for the three data sets described in the previous Section, the mutual information between the features extracted from the first packets of each flow and the corresponding protocol Y. The entropy of the variable Y that represents the application we want to infer from the extracted features, is $H(Y) = \log_2 P$, where P is the number of protocols we consider; in our experiments P is 8 for all data sets. This value of $H(Y)$ is due to the use of the same number of flows for each class, that implies the same weight to each protocol. We do this to be able to derive the "information content" of the features independently from the traffic mix. Using the same weight for each protocol class to achieve such independence is also supported by existing literature, such as [4].

All the values of mutual information in the Figures are lower than $H(Y)$, that equals to 3 bits, representing an upper limit to the mutual information between a feature, or a vector of features, and Y, that is achievable only when the protocol is perfectly predictable. In the following we analyze separately the properties of each feature.

4.1 Round Trip Time

In Figure 1 we show the values of the mutual information $I(rtt_s; Y)$ and $I(rtt_c; Y)$ between the RTT estimations and the protocol Y. Note that these values do not depend on the index i of the packets but are unique for the entire flow.

The mutual information values settle around 0.5-0.7 bits, except for the rtt_s in the border network traces assuming values larger than 1 bit. The additional information carried by the feature rtt_s in the peripheral part of the network is due to the limited number of servers that for a given protocol open connections. For example, a large part of the flows generated by mail services come from a very limited number of mail servers inside the LANs, and therefore with a very homogeneous set of RTT values. In other words, in the cases where the capture device is close to a number of hosts serving traffic through TCP connections (the case of UNIBS and AUCKLAND traces), the rtt_s feature brings more information than in the case when the observation point is in the network's core (NZIX).

4.2 Origin

The binary feature O specifies if the client, that originates a flow, is inside or outside the local network. We determine its value comparing the client IP address with the range of IP addresses of the LAN that we known. It is useful when we consider protocols that have prevalently the same of the two end–hosts (i.e the client or the server) inside the network. For example, in the traffic collected at the border router of our Faculty, the HTTP server is often outside the network, while the SMTP server is often inside. Moving to the center of the network, this information disappears. We can study this feature only in the UNIBS trace, since it is quite difficult to infer it from the AUCKLAND trace because its addresses were anonymized.

4.3 Packet size

As shown in Figure 1 the feature that gives the greater contribution in discriminating the application protocols is the payload size s_i. This confirms that the size of the first

packets in a TCP flow is the most informative feature that can be used to classify application protocols [1, 3]. Our experiments demonstrate that alone, this feature can carry up to around 2/2.5 bits out of 3 of information.

The values of $I(s_i;Y)$ are very similar for the three data sets, indicating that the information contribution of the features associated with the payload size is relatively constant even when the observation point is placed in quite different locations on the Internet. Furthermore, the "amount of information" carried by this feature, and useful in discriminating the application protocols, appears to have remained stable in the last eight years.

Although the specifications of the protocols have not changed in the past eight years, their implementation and usage modes have. In addition, the sets of protocols we consider for the three traces are quite different. Nevertheless, the analysis shows that the information carried by this feature remains constant.

We observe that the trend of $I(s_i;Y)$ as function of the i-th packet is decreasing for all data sets. The reason is the misalignment of the payload content between the packets in the same position i in the packet sequence of the flows generated by the same protocol, mainly due to different protocol implementations and different operations executed with the same protocol in the course of the communication. The presence of retransmissions or out-of-order packets can also affect this feature, but in the traces we analyzed such events amount to only 4.5%, 5.4% and 9.0% of the flows in the first 6 packets for the UNIBS, AUCKLAND and NZIX traces, respectively.

Finally, we have verified experimentally that the mutual information carried by this feature decreases when *empty* TCP packets are included in the analysis, confirming our choice of considering only packets with payloads. Similar trends hold for the features described in the following.

4.4 Inter-arrival time

The mutual information $I(t_i;Y)$ represents the contribution of the inter-arrival time t_i in predicting the application protocol Y. We observe that t_i is less discriminating than the payload size s_i and it decreases significantly moving to the center of the network. We try to explain this behavior evaluating the factors that determine the value of t_i.

The inter-arrival time, measured at the monitoring node, is affected by various elements: some are useful for protocol recognition, others represent only noise and make the feature t_i less expressive. The factors are:

- the *location* of the transmitting hosts with reference to the monitoring node: the response time depends on the route of the packets and on the number of intermediate nodes they pass through,
- the *traffic condition* of the network,
- the *packet forwarding queues* of the routers: quality of service (QoS) mechanisms can give a different priority level to the flows,
- the *transmission direction* of the i-th and $(i-1)$-th packets, representing the features (d_i, d_{i-1}): when both the packets travel in the same direction, they are often the result of TCP segmentation, and they are sent sequentially,
- the time required for the *elaboration* of the received data and for generating the message to send in the i-th packet, e.g. for reading or modifying a large database or for computing cryptographic operations in an authentication phase or for interacting with the user.

The first three aspects, that do not depend on the index of the packets in the flow sequence and that we consider stable for the first packets, affect also the values of the features RTT and O. The feature O, that exists only at the border network, indicates which communication participant is inside the LAN, close to the monitoring point. The packets sent by this end-point have a low latency time due to the vicinity of the observation point to the end hosts.

We show in Figure 1 also the values of the conditional mutual information $I(t_i;Y|RTT,O)$, indicating the additional contribution that t_i provides, due to the packet directions and to the elaboration, for the prediction of Y, instead of considering only the knowledge of Y carried only from the features (RTT,O). We observe that the trend in the three graphs of this quantity is similar, implying that it is not affected from time and space differences of the data sets. Therefore the lower values of $I(t_i;Y)$ for the NZIX data set depends on the first three aspects listed above that in core network provide a lower information on the application protocol. As we have already observed for the feature RTT, in the UNIBS and AUCKLAND data sets the value of t_i is subject to the influence of local aspects, such as the response time of some servers, that help to discriminate the protocols.

4.5 Packet direction

We also compute the mutual information $I(d_i, d_{i-1}; Y)$ between the direction of packets and the application protocol. We prefer to evaluate at the same time the two features d_i and d_{i-1}, rather than the single feature d_i, because the information is mainly associated to the direction of the current packet comparing to the direction of the previous packet. The features d_i and d_{i-1} indicate if the i-th packet travels in the same direction of the previous, or if the endpoints transmitting data changes. We can therefore distinguish when a data transfer is in progress or when the client and the server exchange a short message sequence, that can represent, for example, the early phases of a user authentication.

5. CLASSIFICATION RESULTS

We now show how the properties of the features we studied so far find confirmation in the discriminating capabilities of different statistical classification approaches. We choose two supervised methods, Naïve Bayes and Neural Network used in [3, 4, 5, 6], and two unsupervised ones, K-means and a technique based on Gaussian Mixture Models (GMM), used in [1, 2]. In the unsupervised classifiers the pre-classification information is used only at the end of the training phase to label the achieved clusters of samples.

We show in Figure 2 the average accuracy we obtain in classifying the samples in set T_2 (see Section 2.2), after training each classifier on the set T_1. The value of the average accuracy corresponds to the average portion of flows correctly identified for the eight protocols.

We train the classifiers on different feature sets, considering all the quantities we examined in the previous Section,

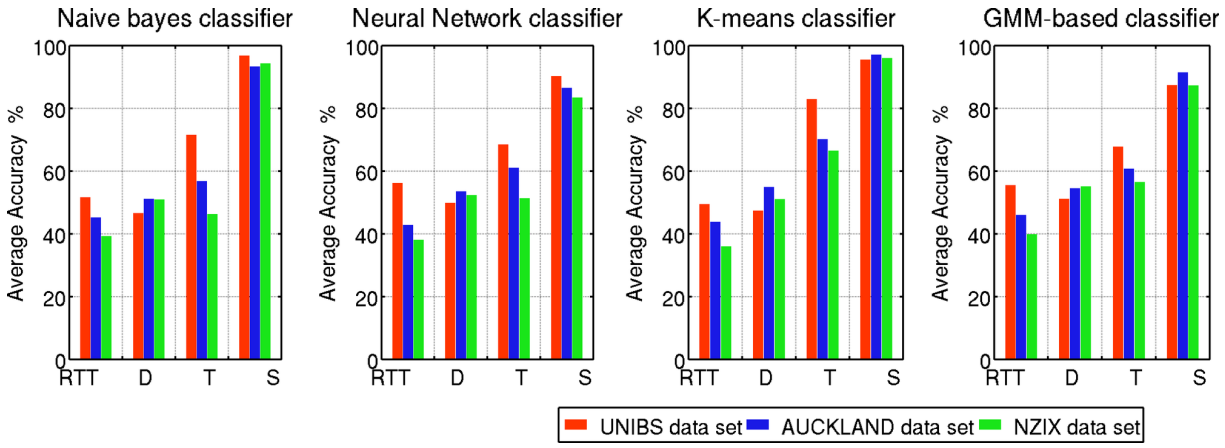

Figure 2: Average accuracy of the classification results using the three data sets we analyzed.

separating the features corresponding to different typologies. Each classifier achieves the higher average accuracy on the feature set composed of the six payload sizes s_i, and the values each classifier reaches are similar for the three data sets. The results obtained using the RTT and T features mimic their information content, higher for the UNIBS and AUCKLAND sets, lower for NZIX.

Finally, it is interesting to note that while the tendencies of each feature set are reflected in the results of all classifiers, there are differences with respect to the capability of each classification technique to extract the information contained in the features. For example, the classifier based on neural networks is less able to extract information from the S features than the Naïve Bayes one, while the two classifiers obtain similar results using the D or T features.

6. CONCLUSIONS

In this paper we presented a statistical analysis for measure the amount of information carried by the most important measurable packet-level features of traffic flows.

The application of this methodology to different traffic traces has shown that the amount of information carried by the features we examined tends to remain relatively constant with respect to both the capture point (from the edge to the core of the Internet), and the capture time (from 2000/01 to 2008).

We find interesting the similarity of the figures of mutual information between packet size and application protocol in the three traces we examined. Such information has remained "stable" over eight years, in two quite different capturing environments and even considering three heterogeneous protocol sets. This indicates that "packet size" seems to be the best single feature any traffic classifier should use. The comparative analysis of four classifiers presented in the previous Section confirms this. Other results also verify what is intuitively clear. For example, inter-arrival times are half as informative when the traffic is captured in the Internet core as opposed to the Internet edge.

We believe that our procedure can be useful in helping to solve some of the problems of statistical traffic classification from the optimization of the feature-sets to the forecast of the accuracy a classifier can obtain when is applied to traffic with different time and space characteristics.

7. REFERENCES

[1] L. Bernaille, R. Teixeira, and K. Salamatian. Early Application Identification. In *Proceedings of CoNEXT'06*, Lisboa, PT, Dec. 2006.

[2] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. *ACM SIGCOMM MineNet Workshop*, Sep. 2006.

[3] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic Classification through Simple Statistical Fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5–16, Jan. 2007.

[4] T. Auld, A.W. Moore, and S.F. Gull. Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, 18:223–239, 2007.

[5] H. Jiang, A. W. Moore, Z. Ge, S. Jin, and J. Wang. Lightweight application classification for network management. In *Proceedings of INM'07*, Kyoto, Aug. 2007.

[6] H. Kim, K. Claffy, M. Fomenkova, D. Barman, and M. Faloutsos. Internet Traffic Classification Demystified: The Myths, Caveats and Best Practices. In *Proceedings of CoNEXT'08*, Madrid, Dec. 2008.

[7] W. Li and A. W. Moore. A Machine Learning Approach for Efficient Traffic Classification. In *Proceedings of IEEE MASCOTS'07*, Istanbul, 2007.

[8] L7 Filter. http://l7-filter.sourceforge.net.

[9] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[10] A. Webb. *Statistical Pattern Recognition*. Wiley, 2nd edition, 2002. ISBN 0-470-84514-7.

[11] N. Schraudolf. Gradient-based manipulation of non-parametric entropy estimates. *IEEE Trans. on Neural Networks*, 15(4):828–837, Jul. 2004.

[12] Tcpdump/Libpcap. http://www.tcpdump.org.

[13] Auckland VI, NLANR/PMA trace repository. http://pma.nlanr.net/Traces/long/auck6.html.

[14] Waikato Internet Traffic Storage (WITS). http://www.wand.net.nz/wits.

[15] NZIX-II, NLANR/PMA trace repository. http://pma.nlanr.net/Traces/long/nzix2.html.