MULTIPLE VECTOR CLASSIFICATION FOR P2P TRAFFIC IDENTIFICATION

F. J. Salcedo-Campos, J. E. Díaz-Verdejo and P. García-Teodoro
CITIC, Dpt. of Signal Theory, Telematics and Communications, University of Granada, Granada, Spain

Keywords: P2P identification, Feature extraction, Flow parameterization, Multiple vector classification.

Abstract: The identification of P2P traffic has become a principal concern for the research community in the last years. Although several P2P traffic identification proposals can be found in the specialized literature, the problem still persists mainly due to obfuscation and privacy matters. This paper presents a flow-based P2P traffic identification scheme which is based on a multiple classification procedure. First, every traffic flow monitored is parameterized by using three different groups of features: time related features, data transfer features and signalling features. After that, a flow identification process is performed for each group of features. Finally, a global identification procedure is carried out by combining the three individual classifications. Promising experimental results have been obtained by using a basic KNN scheme as the classifier. These results provide some insights on the relevance of the group of features considered and demonstrate the validity of our approach to identify P2P traffic in a reliable way, while content inspection is avoided.

1 INTRODUCTION

The wide expansion and increasing popularity of P2P networks and applications gives way to the apparition of some relevant concerns both in traffic engineering and network security. On the one hand, tackling the intensive use of network resources commonly associated to P2P activities represents a challenge for ISPs, that must handle this high volume of traffic with minimal impact on the normal behaviour of the network, while keeping costs under control. On the other hand, the ability to communicate and exchange any kind of information between the so called peers, most of them being anonymous, represents a security risk. This risk first comes from the perspective of the content of the exchanged information or files, which constitutes a security hazard for users through the propagation and execution of viruses, worms, and malware in general. Second, from the networks' infrastructure perspective, as P2P applications are an attack vector that can be used to support other harmful activities as DoS attacks, botnets, and so on.

In this context, it is clear the necessity of differentiating P2P traffic from any other kind of traffic. This is the so called P2P traffic identification problem, which is a specific topic in the more general one of network traffic identification (Callado, 2009). Three main issues arise at this respect:

- **Traffic Parameterization.** Several features have been proposed in the literature to represent network traffic in order to subsequently classify the observed events as belonging to different classes. This way, the data used ranges from the reports of SNMP routers concerning session (connection) statistics (Sen, 2004a) (lower granularity) to TCP headers including the signaling bits and the first bytes of the payload (higher granularity) (Madhukar, 2006). Most approaches just make use of the headers corresponding to RST, SYN and ACK TCP related packets, as there is an underlying assumption about the relevance of the signaling phase for P2P protocols. However, current research is evolving to the use of the overall transport headers for every communication. In some cases, a multiple characterization is proposed. For example, (Chen, 2010) considers the number of ARP packets, the average speed, the average packet size, the proportion of TCP/UDP traffic (IP mode) and the duration (IP-port mode) for a communication.
- **Identification Level.** Once the traffic is parameterized, three different levels are considered in the literature to carry out the identification process (Keralapura, 2010) (Callado, 2009): node-based identification, flow-based identification and packet-based identification. In the first case, the

J. Salcedo-Campos F., E. Díaz-Verdejo J. and García-Teodoro P..
MULTIPLE VECTOR CLASSIFICATION FOR P2P TRAFFIC IDENTIFICATION.
DOI: 10.5220/0003457800050013
In *Proceedings of the International Conference on Data Communication Networking and Optical Communication System* (DCNET-2011), pages 5-13
ISBN: 978-989-8425-69-0

objective is to detect those nodes generating P2P traffic (Xuan-min, 2010). In the flow-based case the goal is to classify each flow as P2P or otherwise. Finally, in packet-based the objective is to classify each individual packet. It is interesting to remark that it is usual to mix those identification levels in different ways. For example, the detection of nodes generating P2P traffic can be dealt with by detecting at least one P2P packet generated by this node. Other approaches like (Keralapura, 2010) use a layered methodology by first detecting nodes and then refining the results to classify the generated flows and, consequently, the associated packets.

- **Identification Process.** Finally, the schemes used to perform the identification itself cover a broad range of techniques. From simple heuristics or indicators (Liu, 2010) (Callado, 2009) (JinSong, 2007) to complex data mining or pattern learning algorithms (Soysal, 2010) (Keralapura, 2010) (Fontenelle, 2007) (Erman, 2007). Moreover, some papers propose to combine several classifiers in a multilayer structure (Yiran, 2010) or in an independent way (Callado, 2010).

Regarding the aforementioned aspects, this paper presents a novel approach for P2P traffic identification, with the following characteristics:

1. A flow-based approach is considered at the identification level. For that, the source and destination ports as well as the source and destination IP addresses, together with transport layer protocol, are used to group individual packets into flows.
2. After that, three main groups of features are obtained for representing every flow at the traffic parameterization stage: transfer related parameters (e.g., packet size), signalling parameters (e.g., number of SYN and ACK TCP packets), and time related parameters (e.g. packet inter-arrival time). This way, a flow is represented by a tuple of three vectors, each one corresponding to a group of features, and some additional parameters related to the flow as the ports or the direction of the first packet. It is important to highlight that none of the features are payload-based.
3. Finally, a triple classification procedure is performed to identify a flow as P2P or otherwise, one for each feature vector. By combining the results of the three classifiers, a final decision is taken to identify a flow as P2P or otherwise. This approach is named as MVC, from *Multiple Vector Classification*.

The items 2) and 3) constitute the specific contributions of this work, they being exposed along the paper as follows. Section 2 describes the general evaluation framework used in experimentation. Section 3 details the feature extraction process in order to detect and represent each traffic flow. After that, a simple KNN-based classifier is used in Section 4 to implement the MVC identification process, from which some experimental results are obtained. Despite the simplicity of the detector used, the results obtained clearly demonstrate the goodness of our approach to identify P2P traffic in a reliable way, and on top of all without requiring payload inspection. Finally, the main contributions of this work and some future research lines are discussed in Section 5.

2 TESTBED

The assessment of identification methods requires the availability of a database of traffic properly classified. This database should be used as the reference to determine the correctness of the results obtained, thus being the ”ground truth”, and should contain enough data so as to be representative. Nevertheless, obtaining a big enough database of labeled traffic is not an easy task, as a manual labeling process is not affordable. Furthermore, the data should be captured in a real network without introducing any artifact, which voids other approaches as injecting known traffic.

Therefore, to evaluate the proposed system we have developed an experimental setup built from two main components: a database of real traffic captured in an academic network, and a tool to automatically classify packets and flows according to their payloads by using Deep Packet Inspection (DPI). This way, the ”ground truth” is built by analyzing and identifying each flow and packet according to this tool under the assumption that DPI is the best currently available method for this and that the number of errors is negligible. This is a common approach in the traffic identification field, the number of packets and flows that DPI is not able to classify being its major limitation.

2.1 DPI Tool

The tool of choice for the classification of traffic is *openDPI* (OpenDPI, 2011), which is derived from the commercial *PACE* product from *ipoque*. The core of openDPI is a software library designed to classify internet traffic according to application protocols. In (Mochalski, 2009) the authors explain that the DPI-based protocol and application classification is achieved using a number of different techniques:

- Pattern matching, by scanning for strings or generic bit and byte patterns anywhere in the

packet, including the payload portion. This way, DPI searches for signatures of known protocols.

- Behavioral analysis, by searching for known behavioral patterns of an application in the monitored traffic. The data used include absolute a relative packet sizes, per-flow data and packet rates, number of flows and new flow rate per application.
- Statistical analysis, by calculating some statistical indicators that can be used to identify transmission types, as mean, median and variation of values used in behavioral analysis and the entropy of a flow.

Therefore, *openDPI* is not a pure-DPI product as it is not only signature-based but also incorporates information from other sources. This way, the classification accuracy is improved (no false classification according to ipoque's claims), although some packets and flows still remains unclassified. This, together with the availability and quality of the signatures, made us to select *openDPI* instead of any other similar product.

For the purposes of our work, we have built a tool based on the *openDPI* library which is able not only to identify the application protocols but also to follow and differentiate the packets in each flow[1]. This way, two classifications are provided: flow-based (each flow is labeled) and packet-based (each packet is also labeled). The tool operates in batch mode and, once the protocol of a flow is known, all the unknown packets in that flow are relabeled as belonging to the identified protocol.

2.2 Traffic Datasets and "Ground Truth"

The traffic database contains data captured during 3 working days for various nodes in a university network. The data acquisition was carried out at a border router in order to be able to monitor all incoming and outcoming traffic for those nodes. Therefore, apart from the boundaries of the caption, flows are captured complete and in both directions. Two datasets S1 and S2 with different groups of nodes are considered to be able to test and validate the method. Table 1 highlights some figures of the database.

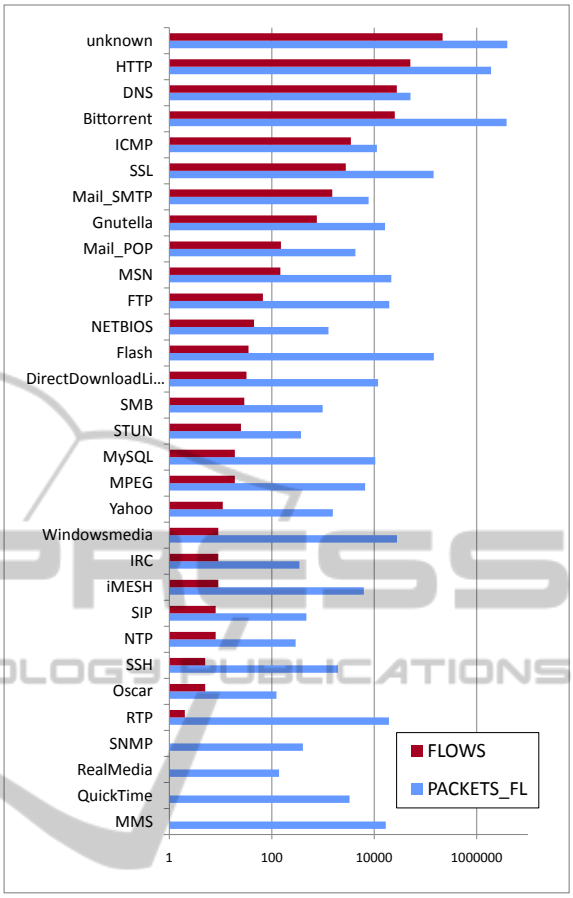

Figure 1: Relative number of instances (flows and packets) of protocols identified in the traffic database.

The results provided by openDPI tool for the database detected a total of 30 different protocols (plus 'unknown') as shown in Fig. 1. The results show that HTTP is the most used protocol while the relative proportion of P2P protocols is lower than expected (only 5-12% of flows). A more detailed analysis of the data shows that only a small number of nodes generate P2P traffic, being videostreaming an important contributor to HTTP traffic (e.g. Youtube traffic).

Most of P2P flows from these nodes are BitTorrent, while Gnutella and others are present in a lower proportion. The rest of the flows (non-P2P) includes mostly usual protocols such as HTTP, DNS, SSL, and mail protocols. The P2P/non-P2P traffic ratio is similar between both sets (see Table 1).

The set S1 will be used to evaluate and tune the system, and S2 to validate the results. In order to increase the confidence of the testing stage, 10-fold stratified cross-validation are used, that is, the ob-

[1]To be able to handle UPD packets, we have generalized the concept of flow through the use of *sessions*. Sessions are considered as defined by the exchange of information associated to a tuple (IP addresses, ports and transport protocol). If the traffic is TCP, a session can be identified as a TCP flow under the assumption that the number of ephemeral ports used by a given IP entity is not greater than 65535 during the observed period. Nevertheless, throughout this paper, we will use the term *flow* to refer to a *session*.

Table 1: Traffic used for classification experiments.

	Flows				
Set	Total	Labeled	P2P flows	Non-P2P flows	Unknown
S1	70797	33524	8091	25433	37273
S2	107860	22645	16005	6640	85215
Total	178657	56169	24096	76475	122488

served flows of S1 were partitioned into 10 random parts with the same number of flows, each presenting the same P2P/non-P2P ratio (Kohavi, 1995). A leave-one-out procedure is applied, taking 9 partitions with correctly labeled examples to train the system and the remaining partition to evaluate it. Thus, 10 different configurations of the partitions for the experiments are used and the results are averaged over the whole set of experiments. The flows were randomly assigned to partitions, in order to enhance the confidence in the results.

2.3 Performance Indicators

In order to compare the results, three well-known measures in the field of classification systems are considered (Gomez, 2002): percentage of true positives (TP), percentage of false positives (FP) and classification accuracy (CA).

Let N_{p2p} and N_{other} denote the total number of P2P and non-P2P messages, respectively. Consider $n_{X\to Y}$ as the number of flows in category X –non-P2P, *other*, or P2P, $p2p$– being classified as belonging to category Y (*other* or $p2p$). The previous measures can be defined in our environment as follows:

1. **TP, True Positives.** The percentage of P2P flows correctly classified as P2P in relation to the total number of P2P flows.

$$TP = \frac{n_{p2p\to p2p}}{N_{p2p}} \cdot 100 \qquad (1)$$

2. **FP, False Positives.** The percentage of non-P2P flows mistakenly classified as P2P in relation to the total number of non-P2P flows.

$$FP = \frac{n_{other\to p2p}}{N_{other}} \cdot 100 \qquad (2)$$

3. **CA, Classification Accuracy.** The percentage of flows correctly classified in relation to the total number of flows.

$$CA = \frac{n_{p2p\to p2p} + n_{other\to other}}{N_{p2p} + N_{other}} \cdot 100 \qquad (3)$$

The ideal system should achieve 100% CA with 100% TP and 0% FP. In order to facilitate the data representation and analysis we use True Negatives (TN), which is an equivalent measure to FP errors. It represents the percentage of non-P2P flows correctly classified as non-P2P in relation to the total number of non-P2P flows. It can be calculated directly from FP rate as $TN = (100 - FP)$.

3 FLOW PARAMETERIZATION

The output of the openDPI tool is a list of the found flows along with their classifications, a list of packets also with their classifications and a pairing list relating flows and packets in each flow. From this information, a parametrization process is applied to obtain a feature vector for each flow, as shown in Table 2, with 61 components. The vectors contain all the information required for their further processing, including an identification label (FLOW_ID), the protocol as detected by *openDPI* and some basic information regarding the flow (flow tuple). IP addresses in a flow are ordered considering them as integers (using network representation) and, therefore, two directions for the packets are considered in the parametrization: UP, for packets traveling from IP_LOW to IP_UPPER, and DOWN for the opposite direction.

The values considered in a parameter vector are basic statistical measures and flow properties, most of them split in total, up and down contributions. Among the parameters are the usual ones included in most netflow-like flow analysis as average packet size, flow duration and number of packets, while at the same time we have included a more detailed description at temporary and signaling level (e.g. interarrival times and number of URG packets).

By analyzing the nature of the parameters, we can consider a feature vector as composed by four parts:

- An identification vector (10 components), which includes all the information required to univocally differentiate each flow and its identification according to *openDPI* (used just to verify the correctness of the classification provided by the proposed system).
- A transfer related vector (24 components), which considers all the parameters related to the number of packets and their sizes.

- A time related vector (10 components), including parameters related to temporary characteristics of the flow, as duration and time between consecutive packets.
- A signaling vector (17 components), that accounts for the number of packets with signaling information and the associated signals.

The values for the parameters are obtained from the list of packets in a flow by analyzing just their sizes, timestamps, TCP flags if any, and the direction of the packets. This way, no inspection of the payload beyond TCP/UDP headers is made, thus preserving the privacy of the users at the application layer. The complexity of the evaluation is low, as only maximum, minimum, count and average values for each parameter are considered.

From the point of view of the classification problem addressed in this work, flow identification parameters, except port numbers, are dismissed, thus resulting in a parameter vector with, at most, 53 parameters.

Table 2: Components of the parameter vector for each flow.

Name	Description
Flow identification	
FLOW_ID	Number of the flow (in the file)
ID_PROT	Detected protocol
IP_LOW	Minor IP address in the session tuple
IP_UPPER	Greater IP address in the session tuple
PORT1	Port associated to minor IP (IP_LOW)
PORT2	Port associated to greater IP (IP_UPPER)
PROT	Transport protocol (TCP/UDP)
DIR	Direction of the first observed packet
FIRST_TIME	Timestamp for the first packet (μs)
LAST_TIME	Timestamp for the last packet (μs)
Transfer related	
NPACKETS	Number of packets in the flow
NPACKETS_UP	Idem UP direction
NPACKETS_DOWN	Idem DOWN direction
PACKETS_SIZE	Total size of the exchanged packets
PACKETS_SIZE_UP	Idem UP
PACKETS_SIZE_DOWN	Idem DOWN
PAYLOAD_SIZE	Total size of payloads
PAYLOAD_SIZE_UP	Idem UP
PAYLOAD_SIZE_DOWN	Idem DOWN
MEAN_PACK_SIZE	Mean size of the packets
MEAN_PACK_SIZE_UP	Idem UP
MEAN_PACK_SIZE_DOWN	Idem DOWN
SHORT_PACKETS	Number of short packets
SHORT_PACKETS_UP	Idem UP
SHORT_PACKETS_DOWN	Idem DOWN
LONG_PACKETS	Number of long packets
LONG_PACKETS_UP	Idem UP
LONG_PACKETS_DOWN	Idem DOWN
MAXLEN	Maximum packet size
MAXLEN_UP	Idem UP
MAXLEN_DOWN	Idem DOWN
MINLEN	Minimum packet size
MINLEN_UP	Idem UP
MINLEN_DOWN	Idem DOWN
Time related	
DURATION	Duration of the flow (μs)
MEAN_INTERAR	Mean time among consecutive packets
MEAN_INTERAR_UP	Idem only for UP packets
MEAN_INTERAR_DOWN	Idem only for DOWN packets
MAX_INTERAR	Max. time among consecutive packets
MAX_INTERAR_UP	Idem only for UP packets
MAX_INTERAR_DOWN	Idem only for DOWN packets
MIN_INTERAR	Min. time among consecutive packets
MIN_INTERAR_UP	Idem only for UP packets
MIN_INTERAR_DOWN	Idem only for DOWN packets
Signaling	
N_SIGNALING	Number of packets with flags
N_SIGNALING_UP	Idem UP
N_SIGNALING_DOWN	Idem DOWN
NACKS	N. of packets with ACK flag active
NFIN	Idem FIN
NSYN	Idem SYN
NRST	Idem RST
NPUSH	Idem PSH
NURG	Idem URG
NECE	Idem ECE
NCWD	Idem CWD
NACK_UP	N. of packets with ACK flag (UP)
NACK_DOWN	Idem DOWN
NFIN_UP	Idem FIN & UP
NFIN_DOWN	Idem FIN & DOWN
NRST_UP	Idem RST & UP
NRST_DOWN	Idem RST & DOWN

4 EXPERIMENTAL RESULTS

The classification of the flows is made by using a KNN classifier and considering different groups of features as previously stated. Therefore, some basics on the use of KNN are described next along with a preliminary analysis on their applicability to the feature vectors considered in this work. From this, the experimental results obtained will be described and analyzed.

4.1 KNN-based Classification

The K-Nearest Neighbors algorithm, or KNN, is a method for classifying objects based on closest training examples in a feature space (Duda, 2001). It is among the simplest machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the most common class among its K nearest neighbors.

Let us suppose that we want to classify the gray triangle as a circle or a square in the space shown in Fig. 2. If $K = 1$, it will be classified as a circle because its most close object is a circle. However, it will be classified as a square if $K = 3$, as two of the three most close objects are squares. The best choice of K depends upon the data. Larger values of K generally reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good K can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when $K = 1$) is called the "nearest neighbor" algorithm, or simply NN.

The training examples are vectors in a multidimensional feature space, each with a class label. Thus, the *training phase* of the algorithm consists only of storing the feature vectors and class labels of

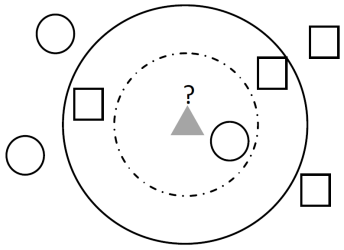

Figure 2: Example of KNN classification: $K = 1$, (big circle in dashed line), $K = 3$ (biggest circle in solid line).

the training samples. For our purposes, as evident, each vector in the feature space is a data transfer related vector, a signaling vector, a time related vector or an overall vector.

In the *classification phase*, K is a user-defined constant, and an observed vector is classified by assigning the most frequent label among the K training samples nearest to that query event. Usually Euclidean distance is used as the distance metric to determine the proximity of two objects in the space. However, other different distance measures are suitable to be applied. In our case, four metrics are used: euclidean, cityblock, cosine and correlation. Thus, the distance between two M-dimensional vectors $X = (x_1, x_2, ..., x_M)$ and $Y = (y_1, y_2, ..., y_M)$ is alternatively calculated as follows:

- Euclidean:

$$d_{euc}(X,Y) = \sqrt{\sum_{i=1}^{M} (x_i^2 - y_i^2)}$$

- Cityblock:

$$d_{cit}(X,Y) = \sum_{i=1}^{M} |x_i - y_i|$$

- Cosine:

$$d_{\cos}(X,Y) = 1 - \frac{\sum_{i=1}^{M} x_i y_i}{\sqrt{\sum_{i=1}^{M} x_i^2} \sqrt{\sum_{i=1}^{M} y_i^2}}$$

- Correlation:

$$d_{cor}(X,Y) = 1 - \frac{1}{M} \sum_{i=1}^{M} \left(\frac{x_i - \bar{x}}{\sigma_x} \right) \left(\frac{y_i - \bar{y}}{\sigma_y} \right)$$

where $\bar{x}$ and $\bar{y}$ are the average components of X and Y, and σ_x and σ_y their standard deviations. The last factor can be calculated through the next expression:

$$\sigma_x = \sqrt{\frac{1}{M-1} \sum_{i=1}^{M} (x_i - \bar{x})^2}$$

4.2 Preliminary Analysis

The parameters list in Table 2 includes different measures over flows: ports used, duration, time between packets, number and length of packets, etc. This variety of parameters implies many statistical differences in terms of average, range or distribution between all of them, especially between parameters from distinct nature. An extreme example is to compare signalization parameters with timing ones. As we can see on Table 3, *DURATION* and *N_SYN* mean values differs in eight or nine orders of magnitude, depending on the traffic type (P2P or non-P2P). Similar differences –$O(6)$ to $O(8)$– can be observed between the statistics of the number of packets per flow (*N_PACKETS*) and *DURATION*. At the same time, the parameter ranges are also very different –from $O(9)$ between *DURATION* and *N_SYN* to $O(3)$ between *N_PACKETS* and *N_SYN*–. These wide differences in ranges and means between parameters types poses a drawback to classification algorithms, especially those based on distance measures as in the KNN case.

One solution could be to re-scale the parameters using a logarithm function (Yuan, 2010), so all of them were distributed in the same value range between 0 and 1. This option has the disadvantage that all the parameters are treated in a similar way, but this is not realistic. We cannot ignore that there are some parameters with a continuous nature, for example all those related with time, and some others with an evident discrete nature, like signaling parameters. This fact suggests us to treat them separately, in order to empower classification techniques.

On the other hand, it is likely that some parameters provide more information than others and even that some parameters do not provide any evidence at all for classification purposes. If all the parameters are treated in a homogeneous way, that is, if all the ranges are somehow normalized, the classification accuracy can be diminished (features without discriminative information will behave as noise for the classifier). Therefore, an approach in which only selected features are evaluated in order to determine their discriminative capabilities when compared against the others becomes very interesting. Therefore, the classifier to evaluate will be composed of a set of classifiers, one per group of features, and the final decision will be made by combining the outputs provided by each individual classifier. This approach, named *Multiple Vector Classification* is similar to that of MVQ that has been successfully applied in speech recognition (Segura, 1994).

The experiments will be designed to address the

improvement of the classification that can be achieved by using this method when compared to a single classifier whose inputs are vectors composed of all the features in Table 2. Therefore, a preliminary experiment with the full feature vector over S1 has been made to be used as the reference system. The results are shown in Fig. 3, where a KNN classifier is used as indicated in Section 4.1 with four different distance metrics: euclidean, cityblock, cosine and correlation. The number of nearest neighbors used in the classification test ranged from $K = 1$ to $K = 10$. The best results were obtained for $K = 4$, which are similar to other reported in the literature. The results do not differ significatively between the considered distances in terms of classification accuracy (CA), because all of them achieve the same true negatives (TN) results. Thus, the differences are mainly based on the TP measure.

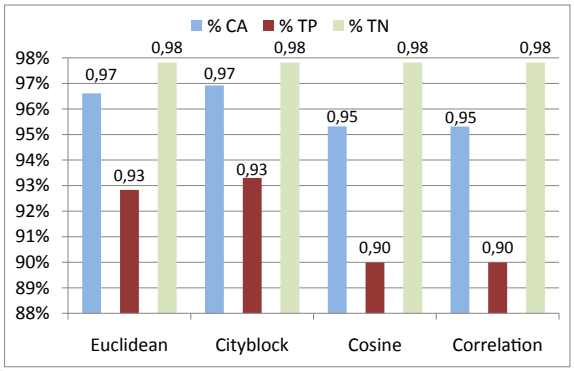

Figure 3: KNN classification results over S1 flows using all the features in a single vector.

4.3 MVC: Identification Results and Analysis

As previously explained, the 53 parameters obtained for representing flows have been split in three sets according to its nature (Table 2):

1. **Time Related.** There exists 10 parameters related to time in the feature vector like the duration of the flows, arrival intervals between packets, etc.
2. **Data Transfer Related.** Include all the parameters indicating volume, like the number of packets of the flow, number of bytes of the packets, etc. There are 24 parameters belonging to this category.
3. **Signaling Related.** There are 19 parameters related to signalization, like the whole number of signaling packets exchanged in the flow, and the number of each kind of signaling packet (ACK, FIN, SYN, URG, etc.).

According to some works in traffic identification –e.g. (Sen, 2004b)–, and despite the use of ephemeral ports on many protocols, port numbers can still be informative to differentiate some protocols. Nevertheless, they are not included in any of the considered set of features. Due to the lack of relationship of port numbers with time or volume parameters and taking into account that they could be considered as signalization at the application layer, we have included port numbers in the signaling category.

Three different MVC schemes have been evaluated with the same distance metrics and values for K as in the reference system in Fig. 4. They all rely on a first triple KNN classification, one for each group of features or vector, and are as follows:

- **Voting.** The flow is assigned to the class provided by the majority of the KNN for each of the three groups of features. This way, a flow will be identified as P2P if at least two of the KNNs classify it as P2P.
- **Nearest KNN.** The flow is assigned to the category given by the KNN providing the lowest distance, independently of the fact that it corresponds to the time related vector, the signaling vector or the data vector.
- **Lowest Aggregate Distance.** The flow is assigned to the category for which the addition of the distances for each vector provided by the corresponding KNN is the lowest one.

Prior to the evaluation of the above MVC schemes, it is necessary to determine again the best K value and distance for classification purposes, as we are now considering the overall features grouped into vectors with different nature. As before, CA, TP or TN can be used to estimate the best K and distance. Taking into account Figure 3, it seems reasonable to select the distance that provides the better TP results because it is the weaker aspect in the four cases. Thus, cityblock is the best distance for time related and signaling vectors, while correlation distance is the best one for the data transfer vector. The number of nearest neighbors is $K = 4$, the same value that was obtained in Section 4.2.

Taking into account these results for the individual KNNs, the evaluation of the three alternate MVC schemes for definitively identifying an observed flow as P2P or otherwise is performed. Figure 4 shows the identification results obtained in comparison with those corresponding to the reference system when a single 53-dimensional vector-based KNN is used.

It is remarkable the TP improvement achieved with the three MVC schemes. Furthermore, both MVC voting and MVC nearest KNN schemes im-

Table 3: Basic statistics of some different flow parameters for S1.

Parameter	Type	P2P			non-P2P		
		Mean	Min	Max	Mean	Min	Max
DURATION	time	$5.58x10^8$	0	$1.69x10^{11}$	$5.04x10^9$	0	$1.72x10^{11}$
N_PACKETS	volume	124.85	1	229507	46.80	1	243444
NSYN	signaling	2.38	0	673	1.67	0	524

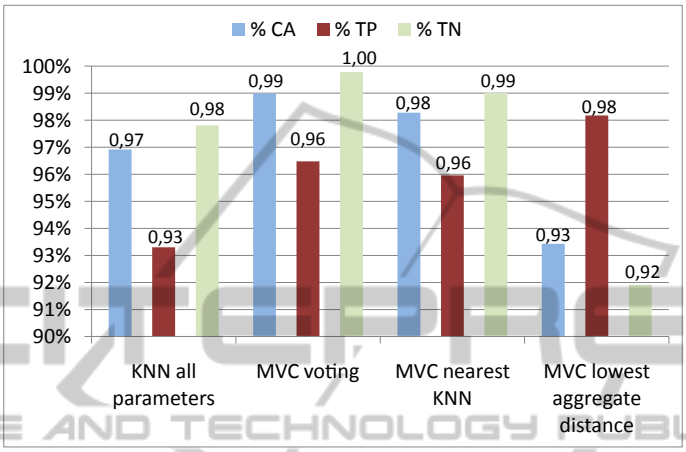

Figure 4: KNN classification results over S1 node flows using different MVC schemes.

prove CA and TN. These figures demonstrate that it is possible to detect more than 96% of P2P traffic with a very low false positives (FP) rate, less than 0.25%. In summary, the MVC schemes outperforms the results obtained by the single KNN classifier with mixed parameters in all the considered measures (CA, TP and TN), with the only exception of MVC with lowest aggregate distance, which just improves the true positives rate.

In order to validate the method, we have tested the best MVC scheme over dataset S2. It is compared with the results obtained by the single KNN classifier with mixed parameters (cityblock distance). Figure 5 shows that the MVC method is still better than using a single KNN with all the parameters together. Therefore, the results over S1 and S2 datasets have shown that the MVC schemes combined with a similarity grouping of parameters is an appropriate method to discriminate between P2P and no-P2P flows.

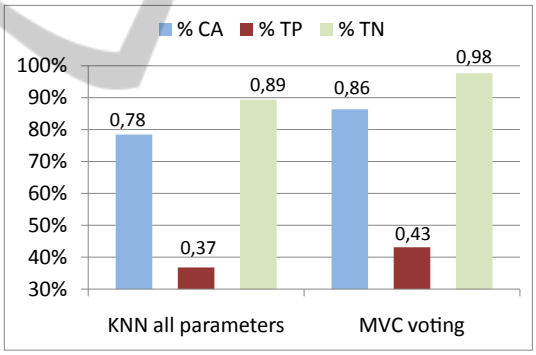

Figure 5: KNN classification results over dataset S2.

5 CONCLUSIONS

A multiple vector classification approach to identify P2P traffic is presented in this paper. It is flow-based, each flow being represented by a tuple of three vectors regarding, respectively, data transfer features, signalling features and time features. A triple classification is subsequently made per flow, one for each feature vector. From this multiple vector classification, a global decision is taken for the flow identification as P2P traffic or not. Although a simple KNN-based classifier is used for implementing the system, the experimental results achieved show the promising nature of our approach in reliably identifying P2P traffic. Furthermore, the identification scheme does not require to access sensible information in packet payloads.

The proposed approach can be improved in some ways. As an example, let us say three. First, a better classifier can be used; for example, SVM has demonstrated to have very good performance in this kind of tasks. Second, each feature vector could be analyzed more in detail in order to reduce its dimensionality

to those more representative characteristics. Third, some alternative combination schemes can be considered in the global identification process.

ACKNOWLEDGEMENTS

This work has been partially supported by Spanish MICINN under project TEC2008-06663-C03-02.

REFERENCES

Callado, A., Kamienski, C., Szabo, G., Gero, B.P., Kelner, J., 2009. "A Survey on Internet Traffic Identification". In IEEE Communications Surveys & Tutorials, vol. 11, n. 3, pp. 37-52.

Callado, A., Kelner, J., Sadok, D., Kamienski, C.A., Fernandes, S., 2010. "Better network traffic identification through the independent combination of techniques". In Journal of Network and Computer Applications, vol. 33, pp. 433-446.

Chen, H., Zhou, X., You, F., Wang, C., 2010. "Study of Double-Characteristics-Based SVM Method for P2P Traffic Identification". In Int. Conference on Networks Security Wireless Communications and Trusted Computing, pp. 202-205.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. "Pattern Classfication". John Wiley & Sons.

Erman, J., Mahanti, A., Arlitt, M., Cohen, I., Williamson, C., 2007. "Offline/RealtimeTtraffic Classification Using Semi-Supervised Learning". In Performance Evaluation, vol. 64, pp. 1194-1213.

Fontenelle, M., Bessa, J., Siqueira, G., Holanda, R., Sousa, J., 2007. "Using Statistical Discriminators and Cluster Analysis to P2P and Attack Traffic Monitoring", In Latin American Network Operations and Management Symposium, pp. 67-76.

Gomez, J.M., Puertas, E., Maa, M.J., 2002. Evaluating cost-sensitive unsolicited bulk email categorization; in Proc. of the ACM Symposium and Applied Computing, ACM Press, pp. 615-620.

JinSong, W., Yan, Z., Qing, W., Gong, W., 2007. "Connection Pattern-based P2P Application Identification Characteristic". In Proc. of Int. Conference on Network and Parallel Computing Workshops, pp. 437-441.

Karagiannis, T., Papagiannaki, K., Foloutsos, M., 2005. "BLINC: Multilevel Traffic Classification in the Dark". In Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 229-240.

Keralapura, R. Nucci, A., Chuah, C., 2010. "A Novel Self-Learning Architecture for P2P Traffic Classification in High Speed Networks". In Computer Networks, vol. 54, pp. 1055-1068.

Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection; in Proc. of the 14th International Joint Conference on Artifcial Intelligence, Montreal, Canada, (1995)

Li, X., Liu, Y., 2010. "A P2P Network Traffic Identification Model Based on Heuristic Rules". In Int. Conference on Computer Application and System Modeling, vol. 5, pp. 177-179.

Madhukar, A., Williamson, C., 2006. "A Longitudinal Study of P2P Traffic Classification". In Proc. of Int. Symposium on Modeling, Analysis and Simulation, pp. 179-188.

Mochalski, K., Schulze, H., 2009. "Deep Packet Inspection. Technology, applications & net neutrality". White Paper. Available at http://www.ipoque.com/resources/white-papers.

OpenDPI, 2011. http://www.opendpi.org

Segura, J.C, Rubio, A.J., Peinado, A.M., García, P., Román, R., 1994. "Multiple VQ Hidden Markov Modelling for Speech Recognition". In Speech Communication, vol. 14, no. 2, pp. 163-170.

Sen, S., Spatscheck, O., Wang, D., 2004. "Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures". In Proc. of the Int. Conference on World Wide Web, pp. 512-521.

Sen, S., Wang, J., 2004. "Analyzing Peer-to-Peer Traffic Across Large Networks". In IEEE/ACM Transactions on Networking, vol. 12, n. 2, pp. 219-232

Soysal, M., Schmidt, E.G., 2010. "Machine Learning Algorithms for Accurate Flow-Based Network Traffic Classification: Evaluation and Comparison". In Performance Evaluation, vol. 67, n. 6, pp. 451-467.

Xuan-min, L., Jiang, P., Ya-jian, Z., 2010. "A New P2P Traffic Identification Model Based on Node Status". In Int. Conference on Management and Service Science, pp. 1-4.

Yiran, G., Suoping, W., 2010. "Traffic Identification Method for Specific P2P Based on Multilayer Tree Combination Classification by BP-LVQ Neural-Network". In Int. Forum on Information Technology and Applications, pp. 34-38.

Yuan, R., Li, Z., Guan, X., Xu, L., 2010. An SVM-based machine learning method for accurate internet traffic classification. Information Systems Frontiers, Springer-Verlag, V. 12, n. 2, pp. 149-156.