J Netw Syst Manage
DOI 10.1007/s10922-013-9293-1

An Autonomic Traffic Classification System for Network Operation and Management

Valentín Carela-Español · Pere Barlet-Ros ·
Oriol Mula-Valls · Josep Solé-Pareta

Received: 20 December 2012 / Revised: 30 September 2013 / Accepted: 4 October 2013

Abstract Traffic classification is an important aspect in network operation and management, but challenging from a research perspective. During the last decade, several works have proposed different methods for traffic classification. Although most proposed methods achieve high accuracy, they present several practical limitations that hinder their actual deployment in production networks. For example, existing methods often require a costly training phase or expensive hardware, while their results have relatively low completeness. In this paper, we address these practical limitations by proposing an autonomic traffic classification system for large networks. Our system combines multiple classification techniques to leverage their advantages and minimize the limitations they present when used alone. Our system can operate with Sampled NetFlow data making it easier to deploy in production networks to assist network operation and management tasks. The main novelty of our system is that it can automatically retrain itself in order to sustain a high classification accuracy along time. We evaluate our solution using a 14-day trace from a large production network and show that our system can sustain an accuracy <96 %, even in presence of sampling, during long periods of time. The proposed system has been deployed in production in the Catalan Research and Education network and it is currently being used by network managers of more than 90 institutions connected to this network.

Keywords Network monitoring · Machine learning · Deep Packet Inspection · Application identification · Self-adaptative system

V. Carela-Español (✉) · P. Barlet-Ros · O. Mula-Valls · J. Solé-Pareta
Dept. Arquitectura de Computadors, UPC BarcelonaTech, Campus Nord, Edif. D6, Jordi Girona,
1-3, 08034 Barcelona, Spain
e-mail: vcarela@ac.upc.edu

Published online: 15 October 2013

1 Introduction

Over the last years, the accurate classification of network traffic has become a key issue for network operation and management. New network applications (e.g., YouTube, Skype), have heavily modified conventional Internet usage. The traditional classification techniques, based on the well-known ports registered by the IANA [1], are no longer valid due to the inaccuracy and incompleteness of their classification results [2, 3]. As a consequence, researchers have proposed a wide range of traffic classification solutions, as shown by the large number of works existing in the literature [2–20]. Although most proposals achieve high accuracy, there is no universal method that is suitable for every possible network scenario. In addition, their deployment in production networks presents practical constraints that proposed methods do not completely address. For example, most machine learning (ML) techniques [6, 11, 21] usually rely on a costly training phase that requires human intervention. As shown by Li et al. [16], these techniques usually require periodic updates in order to adapt to new traffic or new networks. This not only implies the involvement of the network operator, but also a specific knowledge for carrying out the task. Deep Packet Inspection (DPI) techniques need expensive hardware in order to cope with the high data rates of nowadays networks [2, 22–24]. Similarly to ML-based techniques, DPI-based techniques also require periodic updates of the signature base used for the classification. On the other hand, host-behavior-based [4, 19, 20] and IP-based [7, 10] techniques cannot classify a large portion of traffic (i.e., they have low completeness). The IP-based techniques delimit its completeness to the IP addresses seen before. Also, the increment of the Content Delivery Networks has decreased the power of the IP-based techniques. The host-behavior-based techniques usually highly depend on the monitoring point [21] because they need to have a complete perspective of the network in order to find out the behaviors of the different hosts. This hinders their deployment in production networks, with a large number of users and connections. As a result, proposed techniques have enjoyed limited success among network operators and managers, as can be observed by the fact that popular network monitoring systems still use traditional techniques based on port numbers [25, 26].

Unlike the rest of the literature, in this paper we address the traffic classification problem from a practical point of view and propose a realistic solution for network operation and management that can be easily deployed and maintained. First, we develop a traffic classification solution that relies on Sampled NetFlow, a widely extended protocol developed by Cisco to export IP flow information from routers and switches [27]. This complicates the classification [5, 11] but significantly reduces the cost of the solution and allows its rapid deployment in production networks given that most network devices already support NetFlow or one of its variants (e.g., J-Flow, IPFIX). Second, we propose a classification method that combines the advantages of multiple methods, while minimizing the limitations discussed above. Third, we propose an Autonomic Retraining System that can sustain a high classification accuracy during long periods of time. To the best of our knowledge, this is the first solution for traffic classification to provide this feature, which is central for network operation and management, because removes the need of human intervention of previous solutions, which makes the system easier to maintain. Finally, we evaluate the performance of our method using large traffic traces from a production network.

Although some works have also proposed the combination of different classification techniques [9, 12], previous solutions do not support sampling, require packet-level data and cannot automatically adapt the classification model to the changing conditions of the network traffic and applications.

The rest of this paper is organized as follows. The proposed classification technique and the Autonomic Retraining System are described in Sect. 2. Section 3 presents a performance evaluation of our method and analyzes the impact of different retraining policies on its accuracy. Finally, Sect. 4 concludes the paper.

2 Traffic Classification System

This section describes our traffic classification method for Sampled NetFlow and its Autonomic Retraining System. Figure 1 illustrates the architecture of the complete traffic classification system, which is divided into two different parts. Above in Fig. 1, the *classification path* is in charge of classifying the traffic online. In order to achieve this goal we implement a classifier module, called *Application Identifier*. This module (described later in Sect. 2.1) is loaded as a dynamic library in the monitoring tool and it is only fed with NetFlow v5 data. Given that it only needs the information provided by NetFlow, either sampled or not, and does not have to track the flows, the system is swift and lightweight enough to be deployed in large production networks. As a use-case, we integrated this module in the SMARTxAC system [28] and evaluated it with data from a large production university network, as we describe later in Sect. 3.

Below in Fig. 1, the *training path* carries flow-sampled raw traffic to the *Autonomic Retraining System*. This element has two important goals. First, it will provide online information about the current accuracy of the *Application Identifier* that is classifying the traffic online in parallel, as will be described later in Sect. 2.2. Second, it will generate a new classification model when this accuracy falls below a

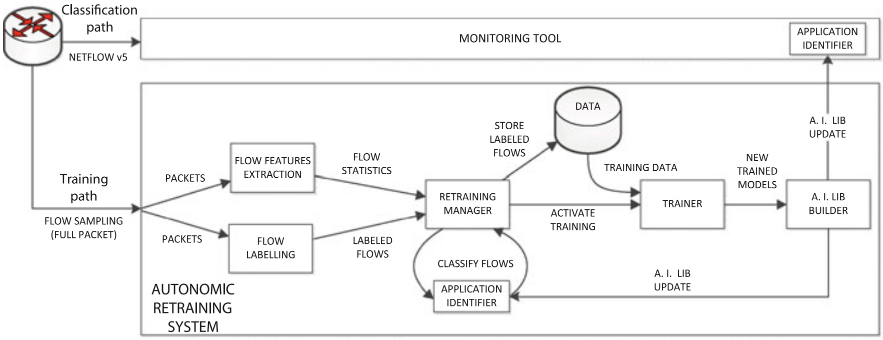

Fig. 1 Application Identifier and Autonomic Retraining System Architecture

threshold, as we describe in Sect. 2.2. Given that the *Application Identifier* is loaded dynamically it could be reloaded at any time. This capability along with the *Autonomic Retraining System* makes our system resilient to traffic variations as, when the accuracy falls, the system will be automatically retrained and the classification model updated.

2.1 The Application Identifier

The *Application Identifier* is the module in charge of the online classification. As aforementioned, it combines different techniques for the sake of exploiting their advantages and reducing their practical limitations. For the reason discussed before, we only consider use methods capable of dealing with Sampled NetFlow data. Furthermore, we need fast classification techniques and methods with a lightweight training phase. Although these constraints complicate the classification, this makes the system easier to deploy and maintain, which are crucial features for network operation and management. The final choice consists of the combination of three techniques of different nature with some improvements especially made to increase their accuracy with Sampled NetFlow data.

Firstly, we use an IP-based technique based on the proposal presented by Mori et al. [10]. Basically, this technique tracks down in an offline phase the IP addresses belonging to famous web sites (e.g., Google, Megavideo). This technique is very accurate, however its completeness has been significantly degraded given the migration of some applications to Content Delivery Networks. This has relegated its use in our system as a technique to be combined with other more complex techniques.

The second method used by the *Application Identifier* is an adaptation of the Service-based classifier described by Yoon et al. [7]. A service is defined as the triplet <IP, Port, Protocol> assigned to a specific application. The list of services is also created in an offline phase using a dataset of labeled flows as follows. In our system, we aggregate all the available flows by their triplet and then, a service is created when a triplet has a minimum number of flows (n) and there is a predominant label ($>m$ %). Unlike in [7], we do not use a time threshold but we require a higher number of flows ($n \geq 5$). We studied offline an efficient configuration of those parameters in order to increase the completeness of the technique while keeping high accuracy. The results have determined that a proper configuration in our setting is selecting $n = 10$ and $m > 95$ %.

The last method used in the *Application Identifier* is a ML technique, namely, the C5.0 decision tree, an optimized successor of the well-known C4.5 presented by Quinlan [29]. To the best of our knowledge, this is the first paper in the field of traffic classification that uses this variant. Nevertheless, several papers have previously highlighted the advantages of its predecessor. Kim et al. [21] and Williams et al. [14] compared different classification techniques showing that C4.5 achieves very high accuracy and classification speed due to its inherent feature discretization [30]. Furthermore, the C5.0 is characterized by having shorter training times compared with its predecessor [31] and with other well-known ML techniques as Support Vectors Machines or Neural Networks [21]. This ability is a key point in

Table 1 Features used by each classification technique

Technique	Features
IP-based	IP addresses
Service-based	IP, Port and Protocol
ML-based	NetFlow v5 features + average packet size + flow time + flow rate + inter-arrival time

our proposal given its importance in the *Autonomic Retraining System*. Because of this we have not applied in our evaluation any improving technique as boosting or bagging. The accuracy obtained by the C5.0 with the default configuration is already very high and the improvement obtained by these techniques was negligible compared with the critical increase of training time. Another important feature of our ML-based technique is its full completeness. As mentioned above, the IP and Service-based techniques have limited completeness given that they rely on IP addresses. However, the ML-based technique allows the *Application Identifier* to classify all the traffic.

It is important to recall that a requirement of our system is that the *Application Identifier* has to work only with Sampled NetFlow traffic. The IP and Service based techniques work properly with Sampled NetFlow because the triplet <IP, Port, Protocol> is not affected by the sampling. However, the ML technique is substantially affected by this constraint, given that NetFlow v5 reduces the amount of features available for the classification and applying sampling considerably impacts on the computation of the features [5, 11]. In order to address this limitation we have implemented the C5.0 ML technique following the recommendations proposed in [11] to improve the classification under Sampled NetFlow, which basically consists of applying sampling to the training phase.

In order to combine the power of the three techniques we combine their classification in a final decision. We give priority to the IP-based technique given the IP addresses have been manually checked. However, given its low completeness, most of the traffic is classified by the Service and ML-based techniques. The distribution of the traffic classified by each technique changes with each retraining, however their contributions are usually around 10 % for the IP-based, 60 % for the Service-based and 30 % for the ML-based technique. Those techniques give their classification decision with a confidence value. The classification decision with highest confidence is selected.

Table 1 presents the features used by each technique, all of them obtained from NetFlow v5 data. This is very important because it allows the *Application Identifier* to be very lightweight and easy to deploy given that it works at flow-level and does not have to compute the features for the classification.

2.2 The Autonomic Retraining System

A common limitation of previous proposals presented in the literature, including the ML techniques proposed in [11], in which our *Application Identifier* is based on, is that they usually require an expensive training phase, which involves manual inspection of a potentially very large number of connections. In order to automate this training phase, we developed a retraining system that does not rely on human supervision. This property together with the ability to classify Sampled NetFlow data, makes our proposal a realistic solution for network operators and managers.

Unlike the *Application Identifier*, the *Autonomic Retraining System* presented in Fig. 1 uses packet-level data as input. This data is labeled with DPI-based techniques and later used to build the base-truth for future retrainings. Applying those techniques online is unfeasible given the high resource consumption. However, our system is able to retrain itself and sustain high accuracy rates along time with very few data. This allows us to apply an aggressive flow sampling rate to the *Autonomic Retraining System* input keeping the system very lightweight and economically feasible for the operation and management of large production networks.

The *Autonomic Retraining System* is divided in three phases. The first one corresponds to the *labelling* and *feature extraction*, the second checks the accuracy and stores the base-truth data and, finally, the last phase retrains and reloads the classifier when it is necessary.

In the *labelling* and *feature extraction* phases, the input data is processed in two different ways. On the one hand, while aggregating the data per flow, a feature extraction is applied to obtain the NetFlow v5 features that would be obtained in the *classification path*. On the other hand, to obtain a reliable base-truth, we use a set of DPI techniques, including PACE, a commercial DPI library provided by ipoque [24]. PACE is known to have high accuracy with low false positive ratio. Moreover, to increase the completeness of the DPI classification we added two extra libraries, OpenDPI [23] and L7-filter [22]. In addition, our system is extensible and allows the addition of new labelling techniques to increase the completeness and accuracy. Based on their relative accuracy, we have given the highest priority to PACE and the lowest priority to L7-Filter. The final label of each flow is selected from the DPI technique with highest priority. An evaluation of the impact of the different DPI techniques used in the *Autonomic Retraining System* is presented in Sect. 3.1.

In the second phase, the *retraining manager* (see Fig. 1) receives the labeled flows together with their NetFlow v5 features. Those that are not labeled as *unknown* are stored for future retrainings. In parallel, the *retraining manager* sends the flows together with their NetFlow v5 features to the *Application Identifier*. This *Application Identifier* is identical to the one that is currently running in the monitoring tool. The *Application Identifier* classifies the flow by obtaining a second label. This label is the same label that the monitoring tool would obtain. By comparing both labels we can compute the actual accuracy of the system. When the accuracy falls under a threshold, we create a new *trainer* in order to build a new classification model. The accuracy in our system is computed from the last flows seen (e.g., 50K in our evaluation). Although the classification is done at the application level, the accuracy is computed aggregating the results at the group level as described in Table 2. Table 2 also presents the traffic mix of the traces used in the evaluation. These traces that are further described in Sect. 3, although collected in a research/university network, are compounded by a heterogeneous mixture of applications.

Table 2 Application groups and traffic mix

Group	Applications	# Flows	
		UPC-II	CESCA
web	HTTP	678,863	17,198,845
dd	E.g., Megaupload, MediaFire	2,168	40,239
multimedia	E.g., Flash, Spotify, Sopcast	20,228	1,126,742
p2p	E.g., Bittorrent, Edonkey	877,383	4,851,103
mail	E.g., IMAP, POP3	19,829	753,075
bulk	E.g., FTP, AFTP	1,798	27,265
voip	E.g., Skype, Viber	411,083	3,385,206
dns	DNS	287,437	15,863,799
chat	E.g., Jabber, MSN Messenger	12,304	196,731
games	E.g., Steam, WoW	2,880	14,437
encryption	E.g., SSL, OpenVPN	71,491	3,440,667
others	E.g., Citrix, VNC	55,829	2,437,664

A *trainer* runs as a separate thread and, using the base-truth dumped in the previous phase, retrains the ML-based and Service-based techniques. The generation of the training dataset is a key point of the retraining system given the important impact it has on the perdurability and accuracy of the models. This process is described in detail in Sect. 2.3. Once the new classification models are built, the new classification library is compiled and dynamically loaded in the *Application Identifiers* that are running on the monitoring tool and the *Autonomic Retraining System* itself. An evaluation of the cost of this process is presented in Sect. 3.2.

2.3 Training Dataset Generation

The proper selection of the instances that compose the training dataset will considerably impact on the quality and perdurability of the new classification models created. This way, we have studied two features to build the training dataset: the retraining policy and the training size of the dataset.

The training size is the number of instances (i.e., labeled flows together with their NetFlow v5 features) that compose the training dataset. We refer to the training size as X. The training size substantially impacts on the training times and the quality of the models. On the one hand, selecting a small training size would produce a system highly reactive to accuracy falls given that the retraining time is shorter. However, the classification models built would be less accurate as they have less information (i.e., instances) to build it. On the other hand, a bigger training size would increase the training times but produce more accurate models.

Regarding the retraining policy we implemented two different policies to perform the retraining. The first approach takes into account the last X labeled flows. However, this approach could be biased to the last traffic received. We refer to it as the *naive retraining policy*. The second approach uses random flows from the last Y days, as follows called *long-term retraining policy*. Although it is totally configurable, for the sake of a fair comparison, we also select a total of X flows proportionally distributed in Y days, where X_i is the number of flows for the day i:

$$X_i = X \cdot \frac{2^{(Y-i)}}{2^Y - 1}$$

Thus, it creates a training data set in which recent days have more weight (i.e., more instances) than older ones.

Section 3.2 evaluates the impact of those features presenting sound conclusions about the best trade-off between accuracy and performance to obtain proper datasets to maintain an accurate online traffic classifier for a network management tool.

3 Evaluation

In this section, we first evaluate the contribution and impact of the different DPI techniques used in the *Autonomic Retraining System*. Then, we evaluate the impact of the policies presented in Sect. 2.3 on the generation of the training dataset. The obtained results are then used to select a proper configuration for the training dataset. Afterwards, we analyze the impact of the *Autonomic Retraining System* on the *Application Identifier*. This evaluation is performed for both sampled and unsampled scenarios. The results show the effectiveness of our system as an autonomous and accurate traffic classifier for large networks.

3.1 Evaluation of Labelling DPI-Based Techniques

DPI-based techniques are scarcely used for online classification given their high resources consumption. However, these techniques are commonly used as an automatic ground-truth generator [8, 11, 21]. In our system, the *Autonomic Retraining System* uses three DPI-based techniques (i.e., PACE, OpenDPI and L7-Filter) to generate the ground-truth online. This is feasible given that the *Autonomic Retraining System* only needs a small sample of the traffic to maintain the *Application Identifier* updated. Samples are selected by applying a high flow sampling rate to the *training path*. This extremely reduces the amount of traffic to be analyzed compared with the whole traffic received by the *classification path*. This type of sampling preserves the entire payload of the flows, allowing DPI-based techniques work properly.

The experiments in this section use the trace *CESCA*. The *CESCA* trace is a fourteen-days packet trace collected on February 2011 in the 10-Gigabit access link of the Anella Científica, which connects the Catalan Research and Education Network with the Spanish Research and Education Network. As described in Sect. 2.2, the *Autonomic Retraining System* only requires a small sample of the traffic to achieve its goal. For this reason, similarly to the flow sampling applied to the *training path*, we applied a 1/400 flow sampling rate. Although the *Autonomic Retraining System* can handle higher flow sampling rates, we applied this one because it was the lowest that allowed us to collect the trace without packet loss in our hardware.

Figure 2 shows the contribution of the different DPI techniques in the base-truth generation. The major contributor in the labelling process is PACE. As Fig. 2a shows, the contribution of OpenDPI and L7-Filter are very low, 0.22 % (0.18 + 0.04 %) and 2.24 % respectively. This is because most of the labels of these techniques match with the labels of PACE and PACE has higher priority (Sect. 2.2). Figure 2b shows that OpenDPI and L7-Filter miss some application labels but match them at group level. This can be seen in the decrease of the PACE percentage. These results also help us to understand the completeness our system would achieve in case we have no access to a commercial labelling tool.

In order to guide the network operator in the selection of an appropriate flow sampling rate for their network, Table 3 presents the consumption of the DPIs techniques by profiling the *Autonomic Retraining System* running in a 3 GHz machine with 4 GB of RAM. Table 3 shows that the average consumption of the different DPI techniques has the same order of magnitude. However, looking at the *standard deviation* and the *maximum (Max)* by flow, L7-Filter behaves totally different than PACE and OpenDPI. This is because L7-Filter has been limited to the first packets and bytes of each flow in order to reduce the false positive ratio [11]. On the other hand, OpenDPI and PACE perform a more thorough examination in order to find out the application label. In more restrictive scenarios, OpenDPI and L7-Filter could be deactivated to improve the performance of the system. However, given that OpenDPI and L7-Filter detects some applications that PACE does not, we have included both DPI techniques in the system. For instance, the 14-days CESCA trace contains 71 million flows (with 1/400 flow sampling applied). Without sampling, 42 µs would be needed in average to process each flow without packet loss (14 days / (71 million flows × 400) = 42 µs/flow). Table 3 shows that only the DPI libraries require 92 µs per flow in average. This shows that a traffic classification system based solely on DPI would not be sustainable in our network scenario and it does not scale well to higher link speeds.

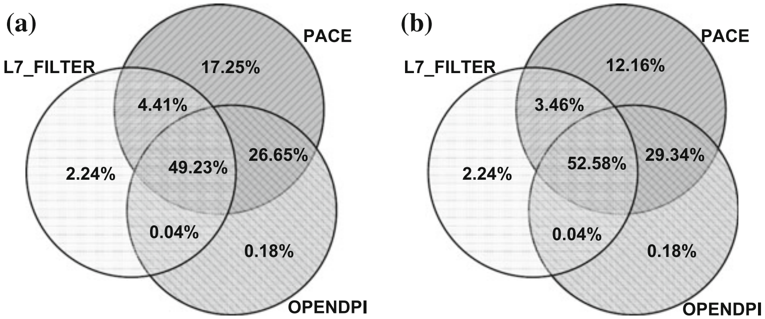

Fig. 2 DPI labelling contribution. **a** By application. **b** By group of application

Table 3 DPI techniques consumption

Metric	DPI techniques		
	L7-Filter	OpenDPI	PACE
Flow			
Avg. (μs/flow)	34.54	25.92	32.36
SD (μs/flow)	41.29	1,419.10	1,721.86
Max (μs)	13,118	1,369,695	1,558,510
Packet			
Avg. (μs/packet)	1.74	1.29	1.66
SD (μs/packet)	10.49	4.31	4.87
Max (μs)	13,118	13,168	12,979

3.2 Training Dataset Evaluation

In this section, we evaluate the impact of the policies presented in Sect. 2.3 in the generation of the training dataset used by the *Autonomic Retraining System*. In all experiments, we use the trace *UPC-II* for the initial offline training and the trace *CESCA* for the evaluation. The trace *UPC-II* is a fifteen-minutes full-payload trace with more than 3 millions of flows. This trace was collected on December 2008 at the Gigabit access link of the UPC BarcelonaTech, which connects about 25 faculties and 40 departments (geographically distributed in 10 campuses) to the Internet through the Spanish Research and Education network (RedIRIS). This link provides Internet access to about 50,000 users. This trace has been also used in [11]. We used different traces for the training and the evaluation in order to show the ability of our system to automatically adapt itself to new scenarios.

In order to asses the quality of the system the *Autonomic Retraining System* uses the accuracy metric. As already mentioned in Sect. 2.2, the *Autonomic Retraining System* computes the accuracy by calculating the number of correctly classified flows from the last flows seen (i.e., 50K in our evaluation). The exact definition of the accuracy metric would be:

$$Accuracy = \frac{\sum_{i=1}^{N}(TP)}{\sum_{i=1}^{N}(TP) + \sum_{i=1}^{N}(FP)}$$

where N: number of categories (i.e., groups of applications). TP (True Positives): The number of correctly identified flows for a specific category. FP (False Positives): The number of falsely identified flows for a specific category.

Although the accuracy is the most popular metric used in the network traffic classification literature it has some limitations. In order to confirm the quality of the *Autonomic Retraining System* we also compute the Kappa coefficient. This metric is considered to be more robust because it takes into account the correct classifications occurring by chance. We computed the Kappa coefficient as explained by Cohen [32]:

Table 4 Long-term policy evaluation

Metric	Training policy (days)			
	5	7	9	11
Avg. accuracy (%)	98.04	98.12	98.07	98.12
Min. accuracy (%)	95.64	95.44	95.44	95.42
# retrainings	126	125	125	125
Avg. training time (s)	229	232	234	242
Cohen's Kappa (k)	0.9635	0.9634	0.9634	0.9633

$$k = \frac{Po - Pe}{1 - Pe}$$

being:

$$Po = \frac{\sum_{i=1}^{N}(TP)}{\sum_{i=1}^{N}(TP) + \sum_{i=1}^{N}(FP)}$$

$$Pe = \sum_{i=1}^{N}(P_{i1} \times P_{i2})$$

where P_{i1}: proportion of apparition of the category i for the observer 1. P_{i2}: proportion of apparition of the category i for the observer 2.

The Kappa coefficient takes values close to 0 if the classification is mainly due to chance agreement. On the other hand, if the classification is due the discriminative power of the classification technique then the values are close to 1.

In order to evaluate the impact of the different retraining policies on the *Autonomic Retraining System* we have performed a study of the impact of the parameter Y in the *long-term retraining policy*. The study evaluates the performance of different values of Y (i.e., 5, 7, 9, 11) with a fixed accuracy threshold (i.e., 98 %) and a fixed training size (i.e., $X = 500\text{K}$). The results of this evaluation, presented in Table 4, show that this parameter has not critic impact on the *Autonomic Retraining System*. However, the values $Y = 7$ and $Y = 11$ achieve the highest accuracies, being $Y = 7$ faster in the training process. As a result, we selected $Y = 7$ for the *long-term retraining policy*. This way, the retraining is performed with flows processed during the last 7 days, allowing the system to cover the traffic of an entire week.

Table 5 presents the results of the evaluation using three different training sizes (i.e., $X = \{100\text{K}, 500\text{K}, 1\text{M}\}$) and two retraining policies (i.e., *naive retraining policy* and *long-term retraining policy*). The evaluation has been performed using a high retraining threshold (i.e., the *Application Identifier* is retrained if the accuracy goes below 98 %) in order to stress the system to perform multiple retrainings by highlighting the differences between the different configurations. Unlike we initially expected, Table 5 shows that the *long-term retraining policy* performs slightly worst than the *naive retraining policy* in terms of accuracy. Moreover, the average training time is shorter for the *naive retraining policy*. This is mainly due the

Table 5 Training dataset evaluation

Training size	Metric	Training policy	
		Long-term policy	Naive policy
100K	Avg. accuracy (%)	97.57	98.00
	Min. accuracy (%)	95.95	97.01
	# retrainings	688	525
	Avg. training time (s)	88	25
	Cohen's Kappa (*k*)	0.9622	0.9567
500K	Avg. accuracy (%)	98.12	98.26
	Min. accuracy	95.44	95.70
	# retrainings	125	108
	Avg. training time ()	232	131
	Cohen's Kappa (*k*)	0.9634	0.9652
1M	Avg. accuracy (%)	98.18	98.26
	Min. accuracy (%)	94.78	94.89
	# retrainings	61	67
	Avg. training time (s)	485	262
	Cohen's Kappa (*k*)	0.9640	0.9650

creation of the dataset that, although it could be optimized for the *long-term retraining policy*, it will be always longer than the *naive retraining policy*. Regarding the training sizes, the option $X = 100$K achieves lower average accuracy than the other training sizes. However, the $X = 100$K training size obtains the highest minimum accuracy and the lowest average training time. This could be interesting if the network demands a fast-recovery system to an accuracy fall. The results comparing $X = 500$K, 1M and the *naive retraining policy* show that these configurations obtain similar average accuracy. However, we have decided to choose $X = 500$K and *naive retraining policy* as the optimum configuration given that it offers slightly better results. Regarding the impact of this policy on the system, the training with a 98 % accuracy threshold only requires 3.93 h compared to the 336 h (14 days) of duration of the whole experiment, which represents only 13 % of the total trace time. If the threshold is lowered up to 96 %, the training time is reduced to 0.54 h (1.8 % of the total trace time).

Although the *Autonomic Retraining System* bases its decisions on the accuracy metric, we have also computed the Kappa coefficient. Table 5 shows that the values of the Kappa coefficient are very close to 1. This result confirms the actual classification power of the *Autonomic Retraining System* showing that its classification is not just due to chance agreement.

3.3 Retraining Evaluation

So far, we have separately studied the performance of the labelling techniques and the impact of the different policies on the training dataset generation. Based on these

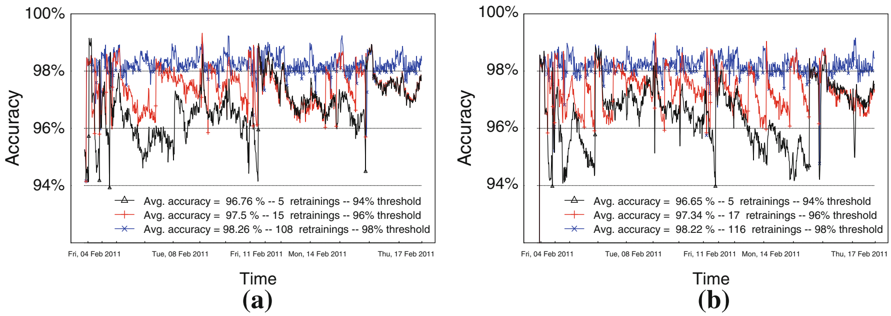

Fig. 3 Impact of the *Autonomic Retraining System* on the *Application Identifier* with the selected configuration (i.e., naive training policy with 500K). **a** Retraining without sampling. **b** Retraining with 1/1,000 sampling rate

results, we selected a final configuration: the *Autonomic Retraining System* uses the three DPI-based techniques (i.e., PACE, OpenDPI and L7-Filter) for the labelling process (Sect. 3.1), 500K flows as training size (i.e., $X = 500\text{K}$) and the *naive retraining policy* (Sect. 3.2). In this section, we evaluate the *Application Identifier* and the impact of the *Autonomic Retraining System* on its accuracy with both sampled and unsampled traffic.

As discussed in Sect. 3.2, we use in all experiments the trace *UPC-II* for the initial training and the trace *CESCA* for the evaluation. It is important to note that the trace *UPC-II* was collected in December 2008 while the trace *CESCA* was collected in February 2011. As a result, the system usually performs an initial retraining to update the initial outdated model. This decision has been taken in order to show the impact of the spatial obsolescence, showing that in order to obtain the most accurate classification model it is crucial to train the system with the traffic of the scenario that it is going to be monitored.

Figure 3a presents the evaluation of the *Application Identifier* when no packet sampling is applied to the traffic. We tested different accuracy thresholds in order to show the behavior of the system depending on the preferences of the network operator. The system maintains the accuracy of 94 % by performing five retrainings during the 14 days. With the 96 % threshold it is able to sustain the accuracy during long periods of time with only 15 retrainings. Using the highest threshold, our method achieves better average accuracy than previous thresholds. However, it is not capable to continuously maintain the 98 % accuracy. Because of this, the *Autonomic Retraining System* is almost continuously updating the classifier. Nevertheless, these continuous retrainings have not any impact on the *Application Identifier* giving that this procedure is done completely apart. Figure 3a also shows the effectiveness of the retrainings, pointed out with cross symbols, that usually produce a substantial increment of accuracy. An interesting result seen by the 94 % threshold is the ability of the system to automatically find a proper classification model. As can be seen at the left part of the Fig. 3a, the system performs three retrainings but finally builds a model that remains stable for about a week.

We have also evaluated the performance of our system when packet sampling is applied in the *classification path*. We perform the experiments with a common

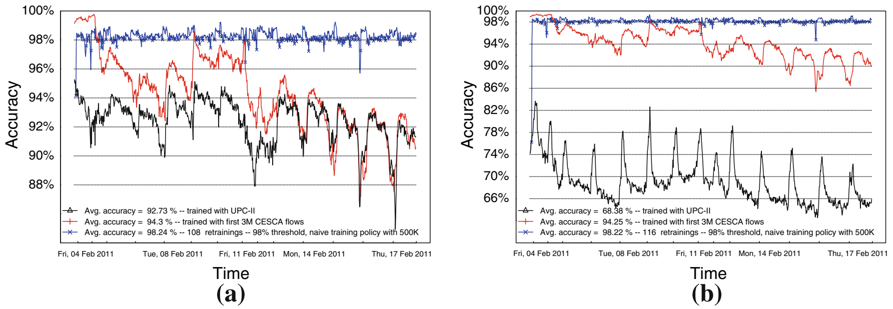

Fig. 4 Comparative of the *Autonomic Retraining System* with other solutions. **a** Comparative without sampling. **b** Comparative with 1/1,000 sampling rate

sampling rate of 1/1,000 using the configuration before described. Figure 3b shows the impact of the retraining in the presence of sampling. The initial low performance showed in Fig. 3b is derived from the fact that we build the initial classification library with the unsampled *UPC-II* trace. As a consequence, the system needs to perform an initial retraining to build a representative model of the current sampled scenario. As aforementioned, this also shows the importance of the spatial obsolescence and justifies the importance of the *Autonomic Retraining System*. Surprisingly, after the initial retraining, the system is able to sustain the same accuracy as the unsampled scenario. The greater decrease of information produced by packet sampling [11] is only reflected in a slight increment in the number of retrainings given that, as described in Sect. 2.1, our techniques has been adapted to deal with it.

Finally, in order to completely understand the influence of the *Autonomic Retraining System*, we have performed two additional experiments that confirm its necessity. The first experiment creates a model with data from one network to classify traffic from another network (i.e., use the trace *UPC-II* to classify *CESCA*). The second experiment creates the model with the traffic of the own network but does not retrain it (i.e., use the *CESCA* trace to train and classify). Giving both trainings can be performed offline we used 3M of flows for both experiments, instead of the 500K of our solution, trying to build the models as accurate and representative as possible. Even so, Fig. 4 show that our solution outperforms these experiments. Figure 4a, that presents the results when no sampling is applied, shows two main outcomes. First, the origin from the data used in the training impacts on the accuracy of the classification (i.e., spatial obsolescence). Even both traces carry traffic from a similar scenario (i.e., educational/research network) there is a substantial difference of accuracy as can be seen in the left part of the figure. The second outcome arises in the right part of the figure where both experiments obtain similar accuracy and this accuracy is gradually decreasing as long as times goes by (i.e., temporal obsolescence). On the other hand, our solution keeps stable during the whole evaluation. Figure 4b, that presents the results when 1/1,000 sampling rate is applied, emphasizes the outcomes previously mentioned. Here, the application of sampling totally deprecates the classification model created with the unsampled traffic of *UPC-II* producing a very inaccurate classification. In both scenarios, the experiments that use *CESCA* for training and classification start with a very high accuracy given that they are classifying the same flows used for the training. Because of that, after the first 3M of flows the accuracy decreases even below than 86 %. However, our solution with its continuous retraining is able to deal with both temporal and spatial obsolescence achieving a stable accuracy beyond 98 %.

3.4 Retraining Evaluation by Institution

As described in Sect. 3.1, the *CESCA* trace was collected in the 10-Gigabit access link of the Anella Científica, which connects the Catalan Research and Education Network with the Spanish Research and Education Network. This link provides connection to Internet to more than 90 institutions. So far, the evaluation has been performed using the complete traffic of the link. This section presents the results of the performance of the *Autonomic Retraining System* with the disaggregated traffic by institution.

Similarly to the previous evaluation we have used 500K flows as training size (i.e., $X = 500$K), the *naive retraining policy* (Sect. 3.2) and the highest accuracy threshold (i.e., 98 %). Two different approaches are used in order to study the performance by institution. First, the *Autonomic Retraining System* uses its normal operation (i.e., using all the traffic and performing the retrainings based on the total accuracy). However, only the accuracy related to the specific institution is presented. Second, the operation of the *Autonomic Retraining System* is changed and, instead of using all the traffic, it only uses the traffic related to the specific institution. Also, the decision of retraining is carried out based on the particular accuracy of the institution and not with the total one. Figure 5 presents the results of this evaluation for three different institutions. The results show the reliability of the *Autonomic Retraining System* for achieving high accuracies with different institutions and scenarios. Although the accuracy is very similar between the three institutions, three different behaviors can be observed. *Institution A* plotted in Fig. 5a has a very volatile accuracy. Even when the model is trained with its own traffic the accuracy is sharply changing, although almost always keeping an accuracy higher than 92 %. On the other side, *Institution C* plotted in Fig. 5c has a more stable accuracy. This is translated into a smaller number of trainings compared with *Institution A*. Finally, the behavior of *Institution B* would be among the other two. These three behaviors plotted in Fig. 5 are the result of different grades of heterogeneity (i.e., *Institution A*) and homogeneity (i.e., *Institution C*) in the traffic of the institutions.

Another interesting output from this evaluation is the impact of the origin of the training data on the accuracy. Figure 5a shows that *Institution A* achieves higher accuracy performing the retrainings with data from the complete link. However, Fig. 5c shows that *Institution C* achieves higher accuracy when the classification model is created with its own data. It is important to note that the amount of retrainings is not comparable between the two approaches. Although the configuration is the same between them, the amount of data available for each approach is

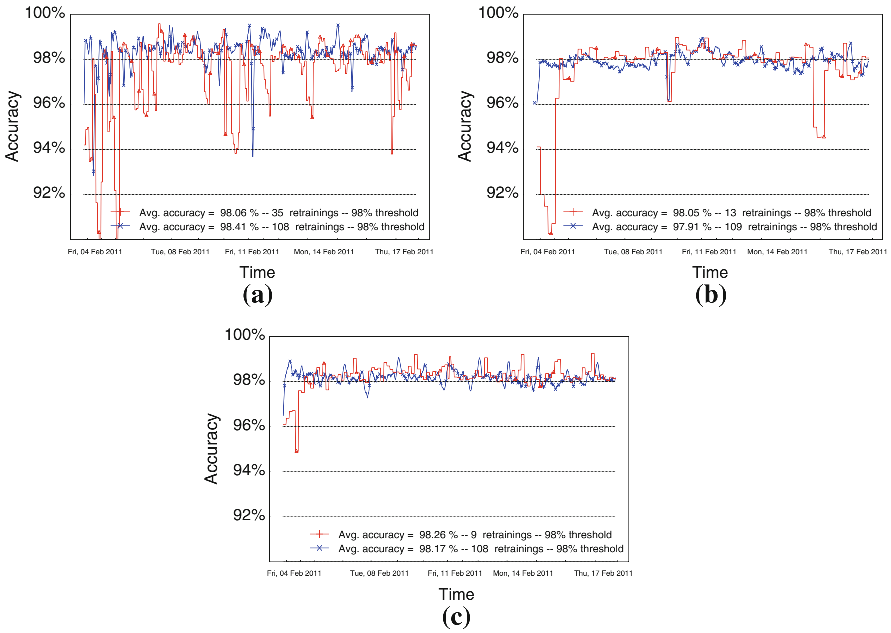

Fig. 5 Comparative of the *Autonomic Retraining System* by institution. **a** Institution A. **b** Institution B. **c** Institution C

different. These two different results regarding the two approaches could be also related to the grade of heterogeneity of the traffic. Training the classification model with traffic from others institutions can help to classify unexpected traffic (e.g., new applications) in networks with heterogeneous traffic.

All these results confirm that the combination of the three techniques and the ability to automatically update the classification model outperform the solutions proposed in the literature for Sampled NetFlow traffic classification [5, 11]. The proposed system has been deployed in production in the Catalan Research and Education network and it is currently being used by network managers of more than 90 institutions connected to this network.

4 Conclusions

In this paper, we presented a realistic solution for traffic classification for network operation and management. Our classification system combines the advantages of three different techniques (i.e., IP-based, Service-based and ML-based) along with an autonomic retraining system that allows it to sustain a high classification accuracy during long periods of time. The retraining system combines multiple DPI techniques and only requires a small sample of the whole traffic to keep the classification system updated.

Our experimental results using a long traffic trace from a large operational network shows that our system can sustain a high accuracy (>96 %) and completeness during weeks even with Sampled NetFlow data. We also evaluated different training policies and studied their impact on the traffic classification technique. From these results we can draw several conclusions:

- The classification models obtained suffer from temporal and spatial obsolescence. Our results in Sect. 3.3 confirm this problem which was already pointed out by Li et al. [16]. Our system addressed this problem by implementing the *Autonomic Retraining System* that is able to automatically update the classification models without human supervision.
- The life of the classification models is not fixed. As indicated by the results in Sect. 3.4, we show that the frequency of retrainings partially depends on the grade of heterogeneity and volatility of the traffic in the network.
- Although several classification techniques have been proposed by the research community, there is no one suitable for all the types of traffic and scenarios. We truly believe that the combination of different techniques is the best approach for properly classifying all the different types of traffic. Our approach, based on three different techniques, is able to achieve very high accuracy and completeness, something that would not be possible if they were not combined.

In summary, we presented a traffic classification system with several features that are particularly appealing for network management: (1) high classification accuracy and completeness, (2) support for NetFlow data, (3) automatic model retraining, and (4) resilience to sampling. These features altogether result in a significant reduction in the cost of deployment, operation and maintenance compared to previous methods based on packet traces and manually-made classification models. The proposed system has been deployed in production in the Catalan Research and Education network and it is currently being used by network managers of more than 90 institutions connected to this network.

Although this work is mostly completed and our system has been already deployed in a production network (CESCA), there are three lines of future work we plan to study. First, the increasing importance of Content Delivery Networks has decremented the power of IP-based classification techniques. Consequently, it would be interesting to study the inclusion of more classification techniques as those based on host-behaviors [4, 19, 20]. Second, we plan to include new techniques [33–35] to reduce the amount of unknown traffic and improve the generation of the ground truth. Finally, we plan to study the viability of using stream-oriented ML techniques, given that its streaming operation seems more suitable for processing the network traffic.

Acknowledgments The authors want to thank ipoque for kindly providing access to their PACE software and Tatsuya Mori for sharing with us the list of IPs presented in [10]. We would also like to thank UPCnet and CESCA for the traffic traces provided for this study. This research was funded by the Spanish Ministry of Economy and Competitiveness under contract TEC2011-27474 (NOMADS project) and by the *Comissionat per a Universitats i Recerca del DIUE de la Generalitat de Catalunya* (Ref. 2009SGR-1140).

References

1. Internet Assigned Numbers Authority (IANA): http://www.iana.org/assignments/port-numbers
2. Moore, A., Papagiannaki, K.: Toward the accurate identification of network applications. In: Proceedings of Passive and Active Measurement Conference (PAM), pp. 41–54 (2005)
3. Dainotti, A., Gargiulo, F., Kuncheva, L., Pescape, A., Sansone, C.: Identification of traffic flows hiding behind tcp port 80. In: IEEE International Conference on Communications (ICC), pp. 1–6 (2009)
4. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: multilevel traffic classification in the dark. In: Proceedings of ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM), pp. 229–240 (2005)
5. Jiang, H., Moore, A., Ge, Z., Jin, S., Wang, J.: Lightweight application classification for network management. In: Proceedings of the ACM SIGCOMM Workshop on Internet Network Management (INM), pp. 299–304 (2007)
6. Nguyen, T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. **10**(4), 56–76 (2008)
7. Yoon, S., Park, J., Park, J., Oh, Y., Kim, M.: Internet application traffic classification using fixed IP-port. Manag. Enabling Future Internet Chang. Bus. New Comput. Serv. **5787**, 21–30 (2009)
8. Carela-Espanol, V., Barlet-Ros, P., Sole-Simo, M., Dainotti, A., de Donato, W., Pescape, A.: K-dimensional trees for continuous traffic classification. In: Proceedings of Traffic Monitoring and Analysis (TMA) pp. 141–155 (2010)
9. Li, J., Zhang, S., Li, C., Yan, J.: Composite lightweight traffic classification system for network management. Int. J. Netw. Manag. **20**(2), 85–105 (2010)
10. Mori, T., Kawahara, R., Hasegawa, H., Shimogawa, S.: Characterizing traffic flows originating from large-scale video sharing services. In: Proceedings of Traffic Monitoring and Analysis (TMA) pp. 17–31 (2010)
11. Carela-Espanol, V., Barlet-Ros, P., Cabellos-Aparicio, A., Sole-Pareta, J.: Analysis of the impact of sampling on NetFlow traffic classification. Comput. Netw. **55**(5), 1083–1099 (2011)
12. Dainotti, A., Pescapé, A., Sansone, C.: Early classification of network traffic through multi-classification. In: Proceedings of Traffic Monitoring and Analysis (TMA) pp. 122–135 (2011)
13. Lee, S., Kim, H., Barman, D., Lee, S., Kim, C., Kwon, T., Choi, Y.: NeTraMark: a network traffic classification benchmark. ACM SIGCOMM Comput. Commun. Rev. **41**(1), 22–30 (2011)
14. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Comput. Commun. Rev. **36**(5), 5–16 (2006)
15. Crotti, M., Gringoli, F.: Traffic classification through simple statistical fingerprinting. ACM SIGCOMM Comput. Commun. Rev. **37**(1), 5–16 (2007)
16. Li, W., Canini, M., Moore, A., Bolla, R.: Efficient application identification and the temporal and spatial stability of classification schema. Comput. Netw. **53**(6), 790–809 (2009)
17. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In: Proceedings of ACM International World Wide Web Conference (WWW), pp. 512–521 (2004)
18. Karagiannis, T., Broido, A., Faloutsos, M.: Transport layer identification of P2P traffic. In: Proceedings of ACM Internet Measurement Conference (IMC), pp. 121–134 (2004)
19. Xu, K., Zhang, Z., Bhattacharyya, S.: Profiling internet backbone traffic: behavior models and applications. In: Proceedings of ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM), pp. 169–180 (2005)
20. Karagiannis, T., Papagiannaki, K., Taft, N., Faloutsos, M.: Profiling the end host. In: Proceedings of Passive and Active Measurement Conference (PAM), pp. 186–196. Springer (2007)
21. Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proceedings of ACM International Conference on Emerging Networking EXperiments and Technologies (CoNEXT), p. 11 (2008)
22. L7-filter, Application Layer Packet Classifier for Linux: http://l7-filter.clearfoundation.com/
23. OpenDPI, The Open Source Deep Packet Inspection Engine: http://www.opendpi.org/
24. PACE, ipoque's Protocol and Application Classification Engine: http://www.ipoque.com/en/products/pace
25. CoralReef Software Suite: http://www.caida.org/tools/measurement/coralreef/

26. Iannaccone, G.: Fast prototyping of network data mining applications. In: Proceedings of Passive and Active Measurement Conference (PAM), pp. 41–50 (2006)
27. Cisco Systems Sampled NetFlow: http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.html
28. Barlet-Ros, P., Sole-Pareta, J., Barrantes, J., Codina, E., Domingo-Pascual, J.: SMARTxAC: a passive monitoring and analysis system for high-speed networks. Campus-Wide Inf. Syst. **23**(4), 283–296 (2006)
29. Quinlan, J.R.: C4.5: Programs for Machine Learning. The Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
30. Lim, Y., Kim, H., Jeong, J., Kim, C., Kwon, T., Choi, Y.: Internet traffic classification demystified: on the sources of the discriminative power. In: Proceedings of ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT), p. 9 (2010)
31. Is See5/C5.0 Better Than C4.5?: http://rulequest.com/see5-comparison.html
32. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. **20**(1), 37–46 (1960)
33. Alcock, S., Nelson, R.: Libprotoident: Traffic Classification Using Lightweight Packet Inspection. Technical Report. University of Waikato (2012). http://www.wand.net.nz/publications/lpireport
34. nDPI, Open and Extensible GPLv3 Deep Packet Inspection Library: http://www.ntop.org/products/ndpi/
35. Zhang, J., Chen, C., Xiang, Y., Zhou, W., Vasilakos, A.: An effective network traffic classification method with unknown flow detection. IEEE Trans. Netw. Serv. Manag. **10**(2), 133–147 (2013)

Author Biographies

Valentín Carela-Español received a B.Sc. degree in Computer Science from the Universitat Politècnica de Catalunya (UPC) in 2007 and a M.Sc. degree in Computer Architecture, Networks and Systems from UPC in 2009. He is currently a Ph.D. student at the Computer Architecture Department from the UPC. His research interests are in the field of traffic analysis and network measurement, focusing on the identification of applications in network traffic.

Pere Barlet-Ros received the M.Sc. and Ph.D. degrees in Computer Science from the Universitat Politcnica de Catalunya (UPC) in 2003 and 2008, respectively. He is currently an associate professor with the Computer Architecture Department of UPC and co-founder of Talaia Networks, a University spin-off that develops innovative network monitoring products. His research interests are in the fields of network monitoring, traffic classification and anomaly detection.

Oriol Mula-Valls received a B.Sc. degree in Computer Science from the Universitat Autnoma de Barcelona (UAB) in 2008 and a M.Sc. degree in Computer Architecture, Networks and Systems from Universitat Politcnica de Catalunya (UPC) in 2011. He was one of the developers of the SMARTxAC monitoring system currently deployed on CESCA. His research interests are in the field of network management and administration.

Josep Solé-Pareta obtained his M.Sc. degree in Telecom Engineering in 1984, and his Ph.D. in Computer Science in 1991, both from the Universitat Politècnica de Catalunya (UPC). Currently he is Full Professor with the Computer Architecture Department of UPC. He did a Postdoc stage (summers of 1993 and 1994) at the Georgia Institute of Technology. His publications include several book chapters and more than 200 papers in relevant research journals (>30), and refereed international conferences. His current research interests are in Nanonetworking Communications, Traffic Monitoring and Analysis and High Speed and Optical Networking, and Energy Efficient Transport Networks. His personal web page is at http://personals.ac.upc.edu/pareta/.