Computer Networks 52 (2008) 2975–2987

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An approach to the identification of network elements composing heterogeneous end-to-end paths ☆,☆☆

Alessio Botta, Antonio Pescapé *, Giorgio Ventre

Dipartimento di Informatica e Sistemistica, University of Napoli "Federico II" Via Claudio, 21-80125 Napoli, Italy

ARTICLE INFO

Article history:
Received 14 December 2007
Received in revised form 24 June 2008
Accepted 24 June 2008
Available online 2 July 2008

Responsible Editor: Prof. A. Marshall

Keywords:
Quality of service parameters
Network measurements
Heterogeneous wired-wireless networks

ABSTRACT

Today's networks are becoming increasingly complex and the ability to effectively and efficiently operate and manage them is ever more challenging. Ways to provide end-to-end Quality of Service have to cope with the increasing heterogeneity of these networks, which is due to the several actors of current network scenarios, from access networks to end-user devices, from protocols to applications and operating systems. Exploiting such heterogeneity, in this paper we present an approach to the identification of each element composing an end-to-end path. Such identification is useful in several situations. For instance, it can improve the performance of adaptive and network-aware applications, it can help intelligent routing approaches, and it can be used in network and service overlay scenarios. Our approach, based on *Bayesian* classifiers, utilizes the measurements and off-line processing of three QoS parameters, that are delay, jitter, and packet loss. To illustrate the capabilities of our proposal, we present the results of a large set of experimentations performed with both different sets of features and different sets of QoS parameters. Using measures related to various time periods, in which the considered paths presented diverse characteristics, we show that a *blind* identification of *network bricks* is possible and that its results present a good degree of generalizability.

1. Introduction

Representing a reality in almost all networking scenarios, heterogeneous networks have gained more and more attention from the research community. In spite of their popularity, quality of service (QoS) analysis and provisioning over such networks is very challenging. In current heterogeneous networks, end-to-end paths are composed of a wide range of different elements using different technologies. End-user devices (EuD), access networks (AN), operating systems (OS), applications, and protocols heighten the heterogeneity of each path. All these variables are normally unknown to network administrators and service providers, making it difficult to offer specific and tailored services (VoIP, Video Streaming, p2p applications, Games, IMS-based applications, etc.) to the users while guaranteeing an acceptable quality.

Therefore, simple, efficient, and accurate end-to-end methods for the identification of the single parts of the whole path can be useful to properly monitor, control, and manage the network. Acquiring this information in a direct way (i.e., asking to the application or to the user) is often not possible and, when possible, an erroneous information may be obtained. For example, in peer-to-peer file sharing applications the user can incorrectly report the bandwidth of his Internet connection in order to be differently considered by his peers. In addition, some techniques

☆ This work has been partially supported by CONTENT EU Network of Excellence (FP6-2006-IST-038423), by OneLab EU Project, and finally from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 216585 (INTERSECTION Project).
☆☆ Preliminary results within the same framework of this work have been recently published in [1].
* Corresponding author. Tel.: +39 0817683856; fax: +39 0817683816.
E-mail addresses: a.botta@unina.it (A. Botta), pescape@unina.it (A. Pescapé), giorgio@unina.it (G. Ventre).

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.06.015

Table 1
Network bricks in our heterogeneous network

Protocol	Sender OS	Receiver OS	Sender AN	Receiver AN	Bitrate	Sender EuD	Receiver EuD
UDP	Linux	Linux	Ethernet	Ethernet	51.2 kbps	Workstation	Workstation
TCP	Windows	Windows	WLAN 802.11b	WLAN 802.11b	102.4 kbps	Desktop PC	Desktop PC
SCTP	Linux familiar	Linux familiar	GPRS	GPRS	204.8 Kbps	Laptop	Laptop
			UMTS	UMTS	409.6 Kbps	Palmtop	Palmtop
			ADSL	ADSL	819.2 Kbps		

for the automatic detection exist but, unfortunately, each of them is able to reveal only a single characteristic of the end-to-end path and no integrated approaches exist.

In this work we propose a framework for the *blind* identification of *network bricks* (see Section 2 for the definition of *network bricks*). With the term *blind* we refer to an identification performed by using parameters different from the network properties we are identifying. More precisely, our identification process is based on QoS measures actively collected at the edges of each considered heterogeneous path. Using the measures as features, we treat the identification as a supervised classification problem. Thanks to well known classification techniques we are able to accurately detect different components of an end-to-end path in an integrated fashion.

To illustrate the capability of the proposed approach we present the results of a large set of experimentations conducted over a real heterogeneous wired-wireless network. In particular, over a number of end-to-end paths, first we present the results of the *network bricks* identification using different sets of features; then, taking into account the relationships among QoS parameters, we present the results obtained with a reduced set of such parameters. Our findings show that the *network bricks* identification is possible and that the accuracy can be sliglty improved by considering a selected subset of statistics. Moreover, they show that discarding a single QoS parameter slightly affects the identification accuracy.

The rest of the paper is organized as follows. Section 2 provides some definitions to clarify our approach, and details the paper contribution. Section 3 contains a short description of the analytical tools we used for the identification. In Section 4 we overview the considered end-to-end paths, measurement methodology, data traces, and statistical tools adopted to determine the features. Section 5 presents the results of the *network bricks* identification. In Section 6 we present and describe related work. Finally, Section 7 ends the paper with conclusions and issues for future research.

2. Problem definition and approach

A *blind* identification of *network bricks* based on QoS metrics allows to identify network elements without physically dealing with them (e.g., identify elements that are physically unreachable).

In this paper we use a number of terms and abbreviations. To help the reader, we provide the following definitions.

Definition 1. *Network brick*. With the term *network brick* we mean a network element or a device component from those shown in Table 1 and graphically represented in Fig. 1. In the following we also simply refer to them as *bricks*.

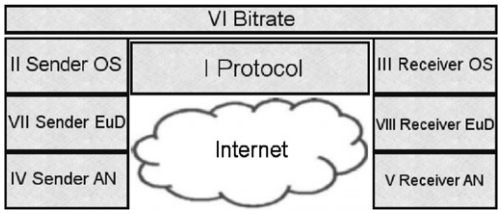

Fig. 1. Abstraction of network scenarios and of the related *network bricks*.

Definition 2. *End-to-end path*. With the term *end-to-end path* we mean a network path composed of *network bricks* (see Fig. 1). Differently from the common literature, our definition of end-to-end path also includes elements like the bitrate, end-user devices, their operating systems, and the transport layer protocol.

Definition 3. *QoS parameter*. With the term *QoS parameter* we mean a parameter among delay, jitter, and packet loss[1]. For each of them some statistics are calculated, which are successively used as features for the classification.

Definition 4. *Feature*. With the term *feature* we mean a statistical indicator calculated over a QoS parameter data trace. In this work we consider the features reported in Table 2 (for more information see Section 4.1). We also divide them in concise and detailed statistics as reported below.

Definition 5. *Concise statistic*. With the term *concise statistic* we mean one of the following: mean, median, minimum, maximum, standard deviation, and inter quantile range.

Definition 6. *Detailed statistic*. With the term *detailed statistic* we mean one of the following: autocorrelation coefficients of Pearson, Spearman, and Kendall evaluated at both lag 2 and 10, and Shannon entropy.

[1] The quality perceived by the users is primarily determined by parameters like packet loss, delay, jitter, and throughput. In our opinion, with the very fast expansion of backbone and core networks, these performance parameters are getting increasingly dominated by the characteristics of the edge of the network. Thus, the observation point for the performance moves from the network core towards the communicating hosts and their access networks.

Table 2
Features evaluated on the samples of the QoS parameters

Concise statistics	Mean, median, standard deviation, minimum, maximum, IQR
Detailed statistics	$r(2)$, $r(10)$, $\tau(2)$, $\tau(10)$, $s(2)$, $s(10)$, entropy

Definition 7. *Identification*. In this work we treat the *network bricks identification* as a supervised classification problem. For this reason the terms *identification* and *classification* will be indifferently used.

Even if the *network bricks* identification constitutes our principal contribution, our work consists of three main parts carried out in three successive phases (see Fig. 2):

- First, we collect several traces of selected QoS parameters over a broad range of heterogeneous network paths. The traces are collected end-to-end with an active measurement approach over the network represented in Fig. 3. More details regarding the adopted measurement approach are provided in Section 4.1.
- Second, after data acquisition and sanitization (see Section 4.1), we calculate some statistics for each considered QoS parameter. The statistics were selected looking at their capability to discriminate the *network bricks* by capturing their peculiar characteristics. In particular, thanks to the analysis we conducted in [2], we verified that the selected statistics actually presented different values for different scenarios and therefore they are the most useful for the identification task.
- Third, using these statistics in a supervised classification algorithm, we identify the *network bricks* composing the end-to-end paths with a high accuracy (i.e., a high percentage of *network bricks* correctly identified). Moreover, we perform different "identification stages" which differ in terms of the feature set they use: *concise*, *detailed*, and *concise + detailed statistics*. Moreover, in a final stage, taking into account the relationships between the considered QoS parameters extensively discussed in literature, we show it is possible to discard all the features related to one QoS parameter without highly affecting the overall identification accuracy. This result appears extremely interesting when only two out of the three parameters are accessible or measurable or when it is necessary to reduce the computation time. In general, this analysis is useful to start the identification process with a sub-set of the statistics, and it provides a clear understanding on the relationships between the features and identification results.

To highlight the significance of our contribution we underline that, to the best of our knowledge, it extends the results present in literature in that: (I) we propose a framework for the *blind* identification of *network bricks*; (II) we show that the *concise* and *detailed statistics* of considered QoS parameters represent a complete and robust set of features; (III) we show that the accuracy can be slightly improved by performing an automatic feature selection; (IV) we present results using real measurements from a number of heterogeneous end-to-end paths; (V) using different sets of measures, collected over different paths, in different time periods, and with different realizations of the same network conditions (e.g., two different UMTS releases), we assess the universality and generalizability of the results.

We would also like to underline that all the data traces we collected are freely available at [3]: this allows other researchers to repeat our experiments and to extend them in order to enrich the research in this field.

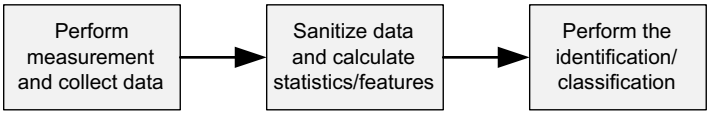

Fig. 2. High level view of the *network bricks* identification life cycle.

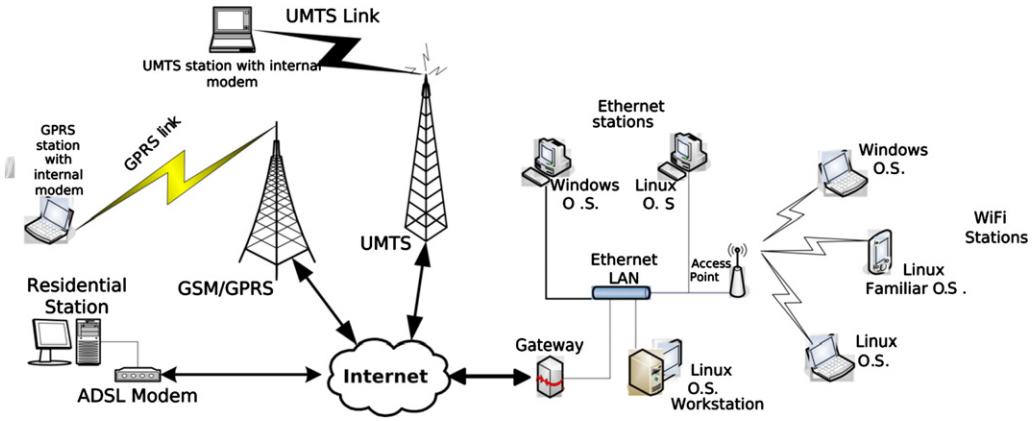

Fig. 3. The real heterogeneous network we use for the measurements.

3. Analytical basis: a brief overview

The idea at the base of the proposed methodology is independent of the particular classification algorithm it uses. In this paper, as a proof of concept, we utilize two classifiers that are widely used in the networking research field. In particular, we use the *Bayesian Network* (BN) classifier and the *Naive Bayesian* (NB) classifier. These classifiers are also chosen because they can be implemented with simple algorithms. This ensures an acceptable processing time [4].

To introduce such classifiers, in this section we provide a brief overview of the *Bayesian Network* and *Naive Bayesian*.

3.1. Bayesian network classifier

Let $S = \{x_1, \ldots, x_n\}$ with $n \geqslant 1$ be a set of variables. A *Bayesian Network* over the set S is a directed acyclic graph (called BN_S) and a set of probability tables $BN_P = \{p(s|pa(s)|s \in S)\}$ in which $pa(s)$ is the set of parents of s in BN_S. A BN represents the chain rule for a joint probability distribution

$$P(S) = \prod_{s \in S} p(s|pa(s)). \tag{1}$$

In general, the classification consists in assigning a set of variables $x = x_1, \ldots, x_n$, called *attribute variables*, to some other variables $y = x_0$, called the *class variable*. The classifier $b : x \to y$ is, therefore, a function that maps each instance of x to the related class y. The classifier learns how to achieve its goal from a data set LS consisting of previously classified points (x,y). For this classifier, the learning stage consists in finding the appropriate BN given a data set LS over S. Once a good network structure is found, the conditional probability tables (for each variable) can be estimated.

In order to perform a classification by using a *Bayesian Network*, it is necessary to simply calculate $\arg\max_y P(y|x)$ using the distribution $P(S)$ represented by the BN. Observing that

$$P(y|x) = P(S)/P(X) \propto P(S) = \prod_{s \in S} p(s|pa(s)) \tag{2}$$

and that all variables in x are known, no complicated inference algorithms are necessary. It is sufficient to calculate Eq. (1) for all class values. For further details refer to [5].

3.2. Naive Bayesian classifier

Let $x = \{x_1, \ldots, x_n\}$ be a data sample representing a realization of $X = \{X_1, \ldots, X_n\}$, and let each random variable X_i be described by m attributes $\{A_1, \ldots, A_m\}$, then $X_i = (A_1^{(i)}, \ldots, A_m^{(i)})^T$ is a random vector. Let $C = \{c_1, \ldots, c_h\}$ be the set of classes of interest. For each observation x_i in x, there is a mapping $C : x \to C$ indicating the membership of instances x_i to a class of interest. Bayesian statistical conclusions about the class c_j, when y is observed, are based on the *a posteriori probability* $p(c_j|y)$. The Bayes rule provides

$$p(c_j|y) = \frac{p(c_j) \cdot p(y|c_j)}{\sum_{c_j} p(c_j) \cdot p(y|c_j)}, \tag{3}$$

where $p(c_j)$ represents the *a priory probability* of class c_j, $p(y|c_j)$ represents the conditional probability of y given c_j, and the denominator is a normalizing factor representing the average probability to observe y. The target of the supervised Bayes classification problem is to estimate $p(y|c_j), j = 1, \ldots, h$ given a training set x. To achieve this goal, the *Naive Bayesian* classifier makes some assumptions on $p(\cdot|c_j)$ such as the independence of A_i, $i = 1, \ldots, m$ and their standard Gaussian behavior. The problem is then reduced to the estimation of the parameters of the Gaussian distribution and $p(c_j)$. Despite its simplicity, the *Naive Bayesian* classifier has been shown to work better than more complex methods and to be able to cope with complex situations [6].

4. Data, tools, and methodology

As introduced in Section 2, our work is carried out in three main steps (see Fig. 2). In this section we provide technical details regarding how these steps are performed.

Fig. 3 shows a scheme of the network on which we apply our measurement and identification approach. The network comprises different heterogeneous wireless/wired connections, as well as different devices and operating systems. The measurements are carried out considering several possible combinations of these variables. In particular, we perform end-to-end throughput, jitter, delay, and packet loss measurements by varying all the possible path characteristics (that are operating system, end-user device, access network, transport protocol, and traffic condition). For more information regarding the network refer to [2].

For the purpose of *network bricks* identification, the selected network scenario exhibits a wide parameter space composed of several variables (see also Table 1) to setup, configure, and analyze. Mixing all these variables we obtain a large number of end-to-end paths. In this paper, to show the applicability of our idea, we present results related to 62 end-to-end paths, with 20 measurements for each path.

The details of the considered paths are reported in Table 3. In such table the number of instances of each *network brick* is reported. As we can see, from all the possible end-to-end paths, in this work we consider those using UDP and not containing Palmtop (as end-user device) or Linux Familiar[2] (as operating system). This allows us to restrict our attention to a reduced set of paths and to include packet loss statistics in our set of features[3]. Despite this, Table 3 allows to understand that the considered scenarios show an high degree of heterogeneity (e.g., different kinds of wired and wireless networks).

4.1. Measurement approach and data traces

We conduct a deep active measurement stage on the heterogeneous network depicted in Fig. 3. This stage allows us to collect data traces related to three QoS parameters (that are jitter, delay, and packet loss). The collected measures are affected by all the characteristics of the het-

[2] Open Source operating system for PDA devices.

[3] Because we work at application level, TCP packet losses are not visible.

Table 3
Heterogeneity tree: number of instances of each *network brick* in the dataset we use for the identification

Protocol				UDP 62			
Sender OS				Linux		Windows	
				36		26	
Receiver OS				Linux		Windows	
				28		34	
Sender AN	ADSL	GPRS	UMTS	Ethernet		WLAN Ad-Hoc	WLAN infrastructure
	3	8	21	12		6	12
Receiver AN	ADSL	GPRS	UMTS	Ethernet		WLAN Ad-Hoc	WLAN infrastructure
	3	15	0	19		6	19
Bitrate [Kbps]				51.2	409.6	819.2	
				18	22	22	
Sender EuD				Workstation	Desktop PC	Laptop	
				27	6	29	
Receiver EuD				Workstation	Desktop PC	Laptop	
				16	3	43	

erogeneous end-to-end paths. This means they reflect the effect of all the *bricks* together. Therefore, as shown in the following, they allow to identify such *bricks*. The measures we use for this work were collected in different time periods. The first one lays between December 2003 and November 2004. A second set of measures was collected between September and November of 2007. A third measurement stage was conducted between January and April of 2008. All the measurements were performed between 9:00 am and 6:00 pm. Using measures related to different time periods allows to assess the generalizability of the results of our analysis. In such measurement stages, over 44 GB of data traces were collected (log file sizes range from about 1 to 100 MB).

The data traces of the QoS parameters are collected with D-ITG [7]. D-ITG is a synthetic traffic generator able to produce a number of traffic patterns by modeling packet size (PS) and inter departure time (IDT) random processes. Therefore, it is capable of generating realistic synthetic traffic while logging data useful to measure the QoS parameters.

The measurements are performed end-to-end, which means that a source of traffic is located at one end of the network path and a sink is located at the other end. Therefore, the jitter, delay, and packet loss measures we obtain are related to the entire path with no information collected by the intermediate nodes.

In order to reduce the number of variables to be considered, we generate only Constant Bitrate (CBR) traffic (i.e., with constant PS and IDT). The measurements are performed by using three traffic conditions named *Low*, *Medium*, and *High Traffic*. These traffic conditions differ in terms of IDT that are 1/100 s, 1/1000 s, and 1/10000 s, respectively. For each IDT, different PS values, ranging from 64 to 1500 Bytes, are used.

In order to simplify the identification problem, our analysis is performed by using the *Low Traffic* condition (IDT = 1/100 s) with PS equal to {64,512,1024} Bytes (see Table 4).

During each measurement, probing traffic is generated for 120 seconds. This duration is chosen to correctly evaluate the performance of the paths, avoiding possible transient phases. Both D-ITG traffic sender (called ITGSend) and receiver (ITGRecv) log some data in a file for each sent and received packet, respectively. A sequence number and two timestamps (sending and receiving time) in the ITG-Recv log file allow to estimate the number of lost packets, the delay, and the jitter. In particular, the samples of such QoS parameters are evaluated by parsing such log file using non overlapping windows of 10 ms. The width of such time windows is the same as the sending period. However, due to the fact that the Internet causes compressions and expansions of the inter packet times, at receiver side we can have more than 1 packet received as well as more than 1 packet lost in each interval. The one way delay is measured in absence of clock synchronization. In spite of this, our identification approach is not influenced by the lack of synchronization. In facts, before evaluating all the statistics related to the delay, for each trace, we subtract the minimum from all the delay values. As a consequence, in each trace, the minimum delay is always equal to 0 and it is not useful to discriminate the paths. Moreover, the mean, median, maximum, and IQR values are not representative of the real values on the network. However, they are still useful to discriminate the different paths and they are therefore utilized as features in the classification process. Moreover, when present the clock skew has been detected and removed [8,9].

Table 4
Parameters of the UDP traffic we use for collecting the QoS parameters samples

IDT	PS	Generated bit rate
1/100 s	64 Bytes	51.2 kbps
1/100 s	512 Bytes	409.6 kbps
1/100 s	1024 Bytes	819.2 kbps

To avoid measure polarization due to external causes of errors, the measurements are interleaved[4]. Moreover, all the collected traces are inspected and sanitized in order to detect and remove samples affected by errors (corrupted log files, network anomalies, etc.).

[4] A time interval has been left between each measurement the length of which depends on the particular network scenario. Moreover, the 20 measurements of each scenario are not taken in sequence but rather interleaved with the measurements related to the other scenarios.

It is worth noting that the GPRS and UMTS connections are provided by three principal Italian telecom operators. Such connections are the same provided to all their customers; for this reason, the data we collect is related to what a common user experiences. It is also worth mentioning that the characteristics of such WWAN connections in Italy have evolved from 2004 to 2008 thanks to the implementation of different UMTS releases (for example the UMTS downlink speed has grown from few hundreds of Kbps to different Mpbs thanks to the introduction of the HSDPA [10]). Using measures related to different standard versions, different telecom operators, and different configurations allows to further assess the generalizability of the obtained results.

4.1.1. Data archives

At [3] we make freely available several archives containing the outcomes of the measurements we made over real networks. Each archive contains a number of text files with the samples of the QoS parameters. The samples are obtained, by adopting the active measurement approach above described, sending probe packets with a rate of 100 pps and a size equal to {64,512,1024} Bytes. More details about the traffic parameters are contained in Table 4.

Each sample is calculated using non-overlapping windows of 10 ms length. The larger is such window, the less number of samples we obtain from each trace and the more packets each sample represents. The chosen length (i.e., 10 ms) constitutes the best trade-off for our traces. At [3] we provide also archives containing samples calculated on a per packet basis.

4.2. Features

Using the samples contained in each QoS parameter trace, we calculate 13 features (see Table 2) in an off-line fashion. They represent a small set of statistics, we divided in *concise* and *detailed*, able to correctly identify the characteristics of the considered *network bricks*. *Detailed statistics* permit to better understand the behavior of the QoS parameters [11,12], as explained in the following.

As for the *concise statistics*, we consider well known parameters like the minimum, maximum, mean, standard deviation, and median values. Also, we consider the inter quantile range (IQR), defined as the difference between the 75th and 25th percentiles. Average and standard deviation are more useful when analyzed along with minimum and maximum values. Moreover, for skewed distributions the IQR and median value are more meaningful than the standard deviation and the average value, respectively. They are indeed less influenced by extreme samples.

As for the *detailed statistics*, we consider the entropy and three correlation coefficients. Generally speaking, the entropy is a measure of the uncertainty of a random variable. Let X be a random variable. The entropy of X, named $H(X)$, is defined as

$$H(X) = -\sum_{x \in X} P(x) \cdot \log_2 P(x). \tag{4}$$

It was used also in [13] to classify network links. To calculate the probabilities we use the Scott rule [14], which allows to choose the width of the bins for the samples.

As regards correlation measures, we use *Pearson*, *Spearman*, and *Kendall* correlation coefficients. The most widely used is the correlation coefficient of Pearson (r)

$$r = \frac{\sum_{i \in [1,n]} (X_i - \overline{X}) \cdot (Y_i - \overline{Y})}{\sqrt{\sum_{i \in [1,n]} (X_i - \overline{X})^2 \cdot \sum_{i \in [1,n]} (Y_i - \overline{Y})^2}}, \tag{5}$$

where $\overline{X}$ and $\overline{Y}$ represent the mean values of two random variables X and Y, respectively. r Ranges from -1 to $+1$. The correlation coefficient of Spearman (s) differs from this one in that the calculations are done after changing the numbers into ranks. Therefore, it can be evaluated by means of Eq. (5) using the ranked data. Spearman correlation coefficient allows to evidence also non-linear relations between the variables.

Both these correlation coefficients are sensitive to outliers and measure the "average dependencies" between random variables. To overcome these limitations and to properly take into account the upper tail dependencies, we also consider Kendall correlation coefficient. Let $(\widetilde{X}, \widetilde{Y})$ be an independent copy of (X,Y). Two observations (x,y) and $(\widetilde{x}, \widetilde{y})$ are then defined as a *concordant pair* if $(x - \widetilde{x}) \cdot (y - \widetilde{y}) > 0$. While they are said to be a *discordant pair* if $(x - \widetilde{x}) \cdot (y - \widetilde{y}) < 0$. We can then define the Kendall's tau (τ) as in Eq. (6) that can be estimated as in Eq. (7).

$$\begin{aligned}\tau(X,Y) = {} & P((X - \widetilde{X}) \cdot (Y - \widetilde{Y}) > 0) \\ & - P((X - \widetilde{X}) \cdot (Y - \widetilde{Y}) < 0),\end{aligned} \tag{6}$$

$$\widetilde{\tau}(X,Y) = \frac{\#\text{concordant pairs} - \#\text{discordant pairs}}{\#\text{pairs}}. \tag{7}$$

Thanks to its properties, Kendall's tau was already used in the study of traffic flow dependences in [15].

In this work all these correlation coefficients are evaluated on the samples of one variable instead of two. Which means that, in spite of Y, a shifted version of X is used (the number of samples of which X is shifted is called *lag*). This allows to evaluate how the samples of the considered variable are dependent (i.e., their autocorrelation). Moreover, they are calculated at both *lag*2 and *lag*10. In a preliminary analysis, in which also the long range dependence properties were considered, such lag values have shown to be the most effective for the purpose of the identification (refer to [2] for more information).

4.3. Identification tool and procedure

Using the features previously calculated, we adopt supervised classification algorithms to identify the *network bricks*. We perform the identification by using version 3.4.9 of WEKA [16], an intuitive and complete software for solving classification and clustering problems.

For training and testing the classifiers, we use the 66% *percentage split* option of WEKA. Which means that out of the 1240 instances we have selected (20 measurements for each of the 62 end-to-end paths), 827 are used for the training and 413 for the tests. The classification is performed for one *brick* at a time. Meaning that, for each *network brick*, we first instruct the classification algorithm

using 827 instances. This phase is called classifier training, and it is followed by the classification phase in which we use the other 413 instances for testing the identification process. The two feature sets are called learning set and test set, respectively. The former represents the attributes the classifier uses to build its model (see Section 3). In such phase, the classifier discovers the peculiar characteristics, in terms of feature values, of each class. In the classification stage, using these characteristics it attributes the elements of the test set to a class. To perform the identification, we instruct the classifier to consider each brick value (e.g., UDP for the protocol *brick*) as a separate class. Therefore, looking at the classification results, we consider a *network brick* instance as correctly identified if the classification algorithm assigns it to the correct class. For each network brick, the identification stage (training + test) takes less than 1 second on a personal computer equipped with an Intel P4@2GHz and 1 GB of RAM.

For assessing the impact of the particular training set on the obtained results, the training/test process is repeated 300 times randomizing, each time, the instances in the training and test sets. In other words, in each test 827 instances are randomly selected from the 1240 in order to train the classifier. Once it is trained, the remaining 413 instances are used for the test. After all the 300 training/test repetitions, the percentages of correctly identified instances (i.e., the identification accuracy) are computed. In Section 5 we report both the mean and the standard deviation of such percentages. Looking at the mean we can verify if the identification process provides accurate results, and how often it makes mistakes. Looking at the standard deviation, instead, we can assess the impact of the particular training set in order to investigate about the generalizability of the results.

5. Experimental results

5.1. Blind identification

In this section we report and discuss the results we obtained. In Table 5, for each *network brick* the average percentage of correctly identified instances is reported together with the standard deviation in parenthesis. Such values are presented for the two considered classifiers. Also, for each *brick* and for each classifier, we report three identification results. Such three results are obtained by using only the *concise statistics*, only the *detailed statistics*, and the complete set of features (*concise* plus *detailed statistics*), respectively. The components of these sets are indicated in Table 2.

As we can see, the two classifiers achieve different performance with the *Naive Bayes* classifier being less accurate. We attribute this behavior to the fact that such classifier assumes the features to be independent, which is not the case here (as discussed in Section 5.3). Moreover, the results from the *Naive Bayes* classifier are much more dependent on the considered set of features. In facts, for some *network bricks* (e.g., Sender EuD) the average percentages obtained with the three sets differ of about 22%. This is consistent with its intrisic simplicity which makes the results more sensitive to the input data. Furthermore, for this classifier, the *detailed statistics* perform better than the *concise statistics*. We attribute this behavior to the fact that the values of the latter set present more skewed distributions. This constrasts with the Gaussian hypothesis of the *Naive Bayes* classifier. For all these reasons the accuracy obtained with such classifier can be as low as 39.1%. This happens for the Bitrate for which we have three possible choices (i.e., 51.2, 409.6, and 819.2 Kbps), and it means that out of the 413 instances used for the test, about 251 (in average) are not ascribed to the correct class but rather to one of the other two. This witnesses that this classifier is not suitable for our aim. For this reason, in the rest of this section, our analysis is focused on the *Bayesian Network* results.

Making no assumptions on the data distribution, the *Bayesian Network* classifier presents higher accuracy and lower differences among the three feature sets. A minor decrease of the performance is noticed when only the *detailed statistics* are used (with the exception of the *Bitrate* and the *Sender EuD*). This is due to the data discretization performed by WEKA *Bayesian Network* implementation [17] which transforms the values of the features from continuous (i.e., $\in \Re$) to discrete (i.e., $\in N$) numbers. Due to the use of a fixed number of bins, this operation allows an easier recognition of the values of skewed distributions, which is the case for the *concise statistics* (Fig. 4 contains the histogram of the relative frequencies of the 1240 realizations of the jitter IQR obtained on the 62 paths).

For the *Sender OS* and the *Receiver OS* the highest accuracy is obtained with the *concise statistics*. In spite of these cases, the *Bayesian Network* achieves the best performance with the complete set of features. For this reason, in the following we discuss only the results related to this kind of identification.

Table 5
Identification results: percentage of identified instances

Brick	Bayesian network (BN), %			Naive Bayes (NB), %		
	Concise	Detailed	Concise + detailed	Concise	Detailed	Concise + detailed
Sender OS	91.1 (3.0)	85.6 (4.6)	90.2 (2.9)	86.8 (3.8)	75.8 (5.4)	86.2 (3.8)
Receiver OS	87.1 (3.3)	80.9 (4.5)	86.8 (3.5)	69.4 (5.3)	75.5 (5.0)	80.2 (4.9)
Sender AN	76.6 (4.9)	69.3 (5.8)	79.7 (5.2)	46.1 (4.9)	61.0 (4.8)	65.4 (4.9)
Receiver AN	72.8 (6.6)	68.3 (6.3)	73.8 (6.1)	47.2 (6.1)	62.0 (5.3)	63.9 (5.4)
Bitrate	58.8 (6.0)	65.3 (5.6)	72.2 (7.4)	39.1 (5.7)	54.6 (5.7)	53.9 (5.9)
Sender EuD	75.3 (5.7)	78.3 (4.9)	78.7 (5.1)	55.3 (5.2)	77.0 (5.2)	63.1 (5.1)
Receiver EuD	84.6 (4.4)	71.6 (5.6)	85.8 (4.1)	51.7 (6.2)	72.9 (5.0)	59.7 (6.9)

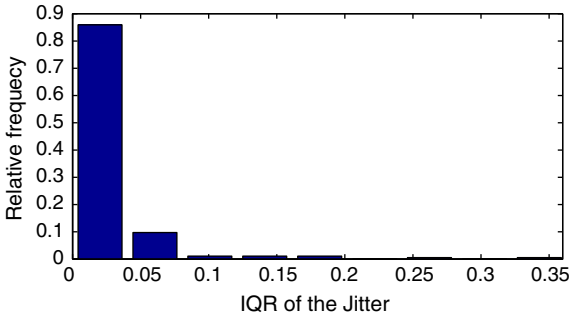

Fig. 4. Relative frequency histogram of the IQR of the jitter.

As one would expect, some *network bricks* are simpler to identify than others. For them, the number of misclassified instances is very low. Digging into numerical details, the best performance is achieved by the *Sender OS*. In this case, the percentage of correctly identified instances is equal to 90.2%. The second most accurately identified *brick* is the *Receiver OS* (accuracy of 86.8%), followed by the *Receiver EuD*, which achieves 85.8%. The remaining *network bricks* (*Sender AN*, *Receiver AN*, *Bitrate*, and *Sender EuD*) achieve approximately the same result (i.e., accuracy $\in[72.2\%, 79.7\%]$).

The obtained results show that the blind identification of *network bricks* is possible and its accuracy is somewhat dependent on the considered *brick*. In overall, it ranges from 72.2% to 90.2%, therefore providing satisfying results. The motivation at the base of such variations is primarly the different number of choices available for the *network bricks*. Indeed, the higher number of candidates are available to the classifier, the more likely it will choose the wrong one.

However, the very good results obtained by the operating systems are also motivated by the different network performance of Linux and Windows, which is reflected by different values of the considered features. From a network-related point of view, the identification of such *brick* is surely of interest for the correct operation and management of the network infrastructure. Different operating systems are often related to different user behaviors and applications. Moreover, while other techniques exist for identifying this *brick* [18,19], we believe our approach is more robust with respect to aspects such as malicious users. For instance, let us assume that a Linux user has bad intents. He may alter the content of specific fields of the packets he generates, in order to emulate a Windows user (e.g., altering the *Time To Live* field of the IP packets, whose default value is differently set by Linux and Windows [19]). While this could cheat a common fingerprinting technique, it would not deceive our approach, which is based on statistical indicators.

A high identification accuracy is also achieved by the end-user devices. This is partly due to the coupling existing between the devices and the access networks. The laptop computers are typically connected through IEEE 802.11 or GPRS/UMTS connections, while the workstations through the Ethernet, and the personal computers often use ADSL or Ethernet connections. Such coupling, which is realistic indeed, is also reflected by our testbed configurations and it caused a higher impact of the *bricks* on the performance which, in turn, resulted in more accurate results. The knowledge of the end-user devices can be very useful in current heterogeneous scenarios. Different devices have different capabilities such as computational power, display size or electrical power availability. This knowledge may be exploited to tailor services to the users and/or to effectively exploit the available resources.

Regarding the access networks, analyzing the obtained results we have verified that the classifier errors are very often due to the confusion of WLAN Infrastructure with the WLAN Ad-Hoc. This also causes a higher standard deviation of the accuracy compared with the other *bricks*. This behavior is due to the very similar characteristics of these two kinds of Access Networks, which are both based on the IEEE 802.11 specifications and achieve very close performance (slight differences between their performance are only due to the presence of the Access Point in the infrastructure mode). However, we can say that the access network, another important element of an end-to-end path, can also be accurately identified. As with the previous one, this information can be surely of interest for users or service providers in order to properly take into account all the characteristics of the communication scenario.

Looking at Table 5, we can also see that, for all the *bricks*, the higher is the average identification accuracy, the lower is its standard deviation. Therefore, the considerations made for the average accuracy, such as the best performing *bricks*, are also valid for the standard deviation (i.e., for the impact of the training set). Moreover, the standard deviation value is less than 5% in average. Which also means that the identification is not heavily affected by a particular training set.

5.2. Blind identification with feature selection

In this section we provide the results we obtained when performing the classification with a reduced set of selected features. This analysis is performed in order to understand how the identification accuracy changes, by using a subset of the features. The best would be to find the same subset for all the *network bricks*. This would mean that some features are not necessary for the identification process, and it may also be exploited for a future online implementation saving preprocessing time by computing only the selected statistics. However, even if the optimal subset is specific for each *network brick*, this information can be used to reduce the time needed for the computations required by the classification algorithms.

The selection of the features is made by using the "*CfsSubsetEval*" algorithm of WEKA with "*Best First Search*" option. This algorithm searches (using the *best first* search) the features that are highly correlated with the class, but loosely correlated with each other. Applying this algorithm to our data we obtain different sets of selected statistics for each *network brick*. The quantity of the *concise statistics* selected out of the 18, and of *detailed statistics* selected out of the 21 is reported in Table 6 together with the average accuracy obtained (the standard deviation is also reported in parenthesis). Such table shows that the accuracy is

Table 6
Results obtained by using a subset of features automatically determined

Brick	# Of selected features		Bayesian network (%)	Naive Bayesian (%)
	Concise	Detailed		
Sender OS	4	2	91.3 (2.8)	87.1 (3.4)
Receiver OS	3	1	88.6 (3.6)	66.1 (6.6)
Sender AN	7	4	80.8 (4.7)	57.5 (5.2)
Receiver AN	6	6	79.2 (5.6)	61.8 (6.2)
Bitrate	3	6	70.3 (5.5)	51.2 (4.5)
Sender EuD	3	8	81.1 (4.6)	66.0 (6.1)
Receiver EuD	4	6	87.0 (4.0)	63.0 (5.8)

slightly increased when compared to the classification performed using all the features (reported in Table 5). Nevertheless, the best performing *network bricks* are the same in both cases. These results witness that a higher accuracy can be obtained by using a reduced set of features composed of both *concise* and *detailed* statistics. However, it is not possible to find a subset which is optimal for all the *network bricks*.

In details, as a first consideration, we can observe a certain correlation between the number of selected features and the achieved accuracy. In particular, excluding the bitrate, the less features are necessary the more accurate are the results. This means that increasing the number of features causes more confusion among the instances.

Moreover, looking in more detail, we can observe that very few features are necessary to identify the sender and receiver operating systems. And, the packet loss statistics are not comprised into such subsets, which means that they are not necessary for the classification. The packet loss features are instead necessary for the identification of the sender and receiver access networks. In particular, for this QoS parameter, the lag-2 correlation coefficient of Pearson is present in both the feature subsets. This is related to the fact that with the injected bitrate (i.e., from 51.2 to 819.2 Kbps) and packet rate (i.e., 100 pps) only the access networks (such as GPRS and UMTS) can be responsible for an observable packet loss. Looking at the results of the end-user devices identification, we have verified that both the selected subsets of features contain the mean, IQR, $r(10)$ (i.e., the lag-10 correlation coefficient of Pearson), and $\tau(2)$ (i.e., the lag-2 correlation coefficient of Kendall) of the delay; while they contain the $r(2)$, $r(10)$ and the entropy of the jitter. Again, in the two subsets no common statistics of the packet loss are present.

Summarizing, we can state that the packet loss statistics are the less frequent in the selected subsets. This behavior is due to the fact that the considered traffic condition does not cause noticeable packet loss, except with narrow-band connections such as the GPRS. This implies that most of the statistics of the packet loss are very close to 0 and, with the exception of the access network, they are not useful to discriminate the *bricks*. Moreover, we have noticed that the median values of the delay and the jitter are the most frequently selected statistics. Such parameters are the most significant for the *network bricks* identification. Other frequently selected statistics are the autocorrelation coefficients of the delay and jitter samples. This last result witnesses that the QoS parameter samples present non-trivial autocorrelation structures as also pointed out in [2]. Finally, we observe that the different subsets of selected features contain different non-common statistics such as the correlation coefficient of Kendall and, in more general terms, both *concise* and *detailed statistics*. This means that set of statistics we consider represents a complete and robust set of features, as anticipated in Section 2.

5.3. Blind identification with the statistics of two QoS parameters

In this section we present the results of the classification performed with the features of two out of the three QoS parameters. We repeated the classification by using the 13 features (*concise* + *detailed* statistics) of only two out of the three previously considered QoS parameters (i.e., the classification uses 26 features instead of 39). This analysis is meant to assess what happens to the identification accuracy when just two QoS parameters are accessible or measurable. Clearly, the investigation is also motivated by the fact that delay, jitter, and packet loss are in some way related to each other. In facts, the relationships between the QoS parameters are extensively studied in literature.

Regarding the packet loss and delay, a number of works have shown the dependencies between them [20–24]. In particular, in [23] the authors analyzed the correlation between packet loss and delay considering that loss events occur in sequence. While in [24] a Hidden Markov Model able to jointly model packet loss and delay over heterogeneous network paths is presented.

Regarding the delay and jitter, a clear analytical dependence exists between them. In particular, in this work the jitter samples are calculated using the following formula

$$j_0 = |d_1 - d_0|, j_1 = |d_2 - d_1|, \ldots, j_k = |d_{k+1} - d_k|, \quad (8)$$

where j_k is the kth jitter sample and d_k is the one way delay experimented by the kth received packet. This formula is compliant with the definition given in [25].

Regarding the packet loss and jitter, due to the fact that they are both related to the delay, it is easy to guess that some relationship between them exists. Studies in literature, especially concerning real time applications and heterogeneous networks, investigate such relation. For real time traffic, packet loss and jitter represent two main causes of problems, therefore they are often jointly monitored and innovative schemes are proposed to cope with both of them at the same time. For instance, in [26] the authors conducted a research to determine the best packet size for VoIP traffic and found that a high traffic load can cause packet loss and jitter, and also that jitter values larger than a threshold may cause packet loss. In a similar context, the authors of [27] found that jitter degrades perceptual quality nearly as much as packet loss does. For heterogeneous scenarios, instead, these two parameters are considered as features to be exploited in order to improve

Table 7
Results of the identification based on two QoS parameters

Brick	Packet loss and jitter (%)		Packet loss and delay (%)		Delay and jitter (%)	
	BN	NB	BN	NB	BN	NB
Sender OS	89.7 (3.3)	85.3 (3.6)	89.9 (3.0)	85.0 (3.7)	90.2 (2.9)	86.7 (3.6)
Receiver OS	84.7 (4.1)	79.6 (6.6)	83.8 (4.0)	81.8 (4.7)	87.5 (3.3)	72.2 (6.3)
Sender AN	70.8 (5.4)	56.4 (5.1)	73.3 (6.2)	55.4 (4.4)	83.1 (4.5)	65.9 (5.4)
Receiver AN	66.7 (7.8)	50.5 (4.5)	72.5 (5.4)	61.7 (5.2)	76.9 (6.1)	59.9 (4.5)
Bitrate	61.9 (6.2)	50.1 (5.4)	68.1 (5.7)	54.9 (6.0)	71.0 (6.6)	54.1 (5.5)
Sender EuD	78.3 (4.8)	62.6 (5.0)	77.5 (5.3)	71.8 (5.0)	77.8 (5.6)	63.0 (4.8)
Receiver EuD	77.5 (6.7)	57.2 (7.2)	84.5 (4.3)	62.2 (6.3)	85.5 (4.1)	58.7 (7.7)

the performance. In [28] Wu and Chen, in order to cope with scarce TCP performance over wireless links, propose a scheme to adapt the sending rate to packet loss and jitter ratios.

According to the results shown in Table 7, the identification based on only two QoS parameters is still possible. With respect to the identification performed by using the features of three QoS parameters (see Table 5), the overall identification accuracy is indeed almost preserved. This confirms that such QoS parameters are dependent and that we can exploit these dependencies in our framework. Table 7 also shows that these dependencies differently influence the *network bricks*.

In particular, we can observe that, for the *Sender OS* and *Receiver OS network bricks*, the identification results for the three QoS parameter pairs slightly differ. This means that the classifier gives almost the same importance to the three parameters. In general, for all the *bricks*, we can state that the average accuracy decreases mainly when the delay features are not taken into account. An accuracy decrease is noticed also when the jitter features are discarded. This means that the least important of the three parameters is the packet loss. This was also confirmed by the previous analysis and it is due to the fact that the considered bitrate and packet rate do not cause significant losses on the network paths. While this can be seen as a drawback of the chosen traffic rate, it is worth underlining that a higher amount of injected traffic would mean more intrusive measurements, which is not desirable in a real operational network.

Concluding, we can state that with the current implementation of the identification process, there is a trade-off between the number of useful statistics and the intrusiveness of the related measurement process. However, we can leave the final decision to the users according to their application requirements.

6. Related work

Inferring network properties from end-to-end measurements represents an important and challenging task. Our approach falls in the general field of edge-assisted network management and control. Previous work has devoted considerable attention to the use of (active or passive) measurements from end-hosts both to infer network properties or performance and for network management. In particular, active measurements can be used for real time performance assessment and diagnosis [29] as well as for inferring network properties [30,31]. Whereas, passive measurements (e.g., routing updates) can be used to detect, localize, and diagnose problems with path performance or Internet routing [32,33]. More recently, a new framework for Internet management and control – called 4D (decision, dissemination, discovery and data) – has been proposed [34]. In the 4D framework, traffic control is moved from routers and switches to end-to-end mechanisms, which rely on packet delay and loss information to adjust traffic intensity. In addition, if multiple paths are available, end-to-end loss/delay information can be used to optimally route traffic from source to destination [35,36]. To pursue these goals new architectures and approaches, often based on the peer-to-peer paradigm, are proposed for collecting and sharing network measures among end hosts [37–40].

Although these research works use interesting techniques and provide useful results, when compared to our proposal they differ significantly in how the data measured at the end hosts is used. Our approach may help end-hosts (or intermediate systems) to identify pieces of networks, opening the way to a wide variety of research studies that aim to provide greater control over the entire end-to-end network paths. Some scenarios could be the following: adaptive and network-aware applications [41–44], reactive and intelligent routing [45,46], overlay networks [47–50], source-routing [51].

Regardless of the final target of our approach, the following represent the closest related works. In [52], the authors propose a passive approach to detect bottlenecks in network paths. The work in [53] presents results on detecting shared congestion of flows by means of end-to-end measurements. The authors of [54] propose inference techniques to estimate the loss rates of network links. Such techniques are based on measures collected on a server. In [55 and 56], approaches aiming to estimate links capacity along a path via end-to-end measurements are presented.

Taking into account the final target of the identification of path elements, our approach is very similar to that presented in [57]. In this work an iterative *Bayesian* technique, based on passive measurements, is used for identifying 802.11 traffic. Differently from [57], we use an active approach to collect our measures. Such an approach has been used also in [13]. In this work, the authors classify the access networks in three classes and show how it is possible to recognize the class elements using the outcomes of an active probing tool. However, differently from these last two works, we are able to identify many elements instead of the access networks only. Finally, several works use TCP/

IP fingerprinting to detect host characteristics. As an example, [18] uses a Bayesian classifier to passively detect the host operating system, whereas in [19] techniques for OS fingerprinting are presented. Compared to the previous, our proposal provides a more complete and integrated approach: using the method described in this paper we are indeed able to identify all the components of an end-to-end path (from access networks to end-user devices, from protocols to end-user operating systems) in a single stage.

As for the limitations, our approach does not explicitly take into account all the issues at the base of developing a software platform that works in an online fashion. The development of such tool requires, in fact, several aspects to be considered (e.g., intrusiveness of the measurement process, scalability, etc.). However, at present we would rather assess the feasibility of identifying end-to-end path components by looking at QoS parameter statistics. More precisely, our approach aims at providing some experimental basis to show that a *blind* identification of different *network bricks* is possible and that our approach is general enough to be applied in several network scenarios.

7. Conclusions and issues for research

The automated discovery of network elements (in this paper we called them *network bricks*) is of great importance for the network administrators in order to get a better knowledge of their networks and, consequently, for improving the network management, configuration and control activities (e.g., intelligent routing, networks and services overlay scenarios, etc.). Nevertheless, it can be utilized by the end-users to efficiently tune the parameters of their adaptive applications. For instance, in a general end-to-end communication scenario, the knowledge of the access network at receiver side (e.g., ADSL or UMTS) helps in setting up efficient control and management activities (e.g., peer selection in peer-to-peer applications, rate adaption in streaming applications, etc.).

In this paper we introduced the problem of (*blind*) identification of *network bricks* over heterogeneous wired-wireless networks. The presented results confirm that our idea of *blind* identification of *network bricks* is feasible. Considering the large variety and heterogeneity of the utilized measures we have also shown that results of the identification have a high degree of universality. We presented results using different sets of features and we showed that a complete set of statistical features performs better than a partial set in overall. In addition, over the complete set (*concise* plus *detailed*), automatic selection techniques permit to reduce the number of features to be considered for the identification, without affecting the accuracy. Finally, we showed that a *blind* identification is possible also with a smaller set of QoS parameters. It represents a *per se* result, also useful when only a limited set of QoS parameters is available (or measurable).

As for the ongoing work, we can divide it into three main categories:

- *increasing the number of available data sets*: we are performing additional measurements over new heterogeneous paths (comprising real and operational heterogeneous wired-wireless networks [58] as well as geographical heterogeneous testbeds [59]). In these new measurements we are also considering both other typologies of traffic sources (VBR, bursty traffic, etc.), more similar to Internet traffic, and other transport protocols (e.g., Datagram Congestion Control Protocol[5]). This will allow to further improve the number of considered *bricks*.
- *looking at the online implementations of the identification process*: as said in the end of Section 6, a limitation of our current implementation is related to the fact that it operates in an offline fashion. It requires, in fact, the data traces to be previously collected and preprocessed. In the present work we were interested in testing the applicability and suitability of our idea of identification. Therefore, we have still to investigate the implications related to developing our approach in an online fashion. This change of modus operandi requires several aspects to be investigated (e.g., intrusiveness of the measurement process, scalability, etc.).
- *working on new identification/classification algorithms*: we are both considering other identification/classification algorithms and multi-classifier approaches. As for the former, we are testing a large number of classifiers to better understand if more appropriate algorithms exist. For testing new algorithms we are also considering what we learned in this study regarding the relations among the *bricks* and among the QoS parameters. As regards the latter, we are considering more classifiers working at the same time and using approaches coming from the decision theory such as Dempster-Shafer theory [60] and Behavior Knowledge Space (BKS) method [61].

References

[1] A. Botta, A. Pescapé, G. Ventre, Identification of network bricks in heterogeneous scenarios, in: Proceedings of the First IEEE LCN Workshop on Network Measurements, Tampa, Florida, 2006.
[2] A. Botta, A. Pescapé, G. Ventre, On the statistics of qos parameters over heterogeneous networks, in: Proceedings of IFIP Networking 2006 Workshop Towards the QoS Internet (To-QoS'2006), Coimbra, Portugal, 2006.
[3] <http://www.grid.unina.it/Traffic>, 2008.
[4] A.W. Moore, D. Zuev, Internet traffic classification using Bayesian analysis techniques, in: Proceedings of ACM SIGMETRICS 2005, Banff, 2005, pp. 50–60.
[5] Remco R. Bouckaert, Bayesian networks in Weka, Technical Report 14/2004, Computer Science Department, University of Waikato, 2004.
[6] I.H. Witten, E. Frank, Data Mining, Morgan Kaufman Publishers, 2000.
[7] <http://www.grid.unina.it/software/ITG/>, 2008.
[8] Sue B. Moon, P. Skelly, D. Towsley, Estimation and removal of clock skew from network delay measurements, in: Proceedings of IEEE INFOCOM 1999, vol. 1, New York, NY, March 1999, pp. 227–234.
[9] L. Zhang, Z. Liu, and C.H. Xia, Clock synchronization algorithms for network measurements, in: Proceedings of IEEE INFOCOM 2002, 2002, pp. 160–169.
[10] <http://www.umtsworld.com/technology/hsdpa.htm>, 2008.
[11] Q. Li, D.L. Mills, On the long-range dependence of packet round-trip delays in Internet, in: Proceedings of IEEE International Conference on Communication 98, vol. 2, 1998, pp. 1185–1191.

[5] We do not consider SCTP (Stream Control Transmission Protocol) for the same reason of TCP.

[12] M.S. Borella, S. Uludag, G.B. Brewster, I. Sidhu, Self-similarity of Internet packet delay, in: Proceedings of IEEE International Conference on Communication 97, vol. 1, June 1997, pp. 513–517.
[13] W. Wei, B. Wang, C. Zhang, J. Kurose, D. Towsley, Classification of access network types: ethernet, wireless LAN, ADSL, cable modem or dialup? in: Proceedings of IEEE INFOCOM 2005, Miami, March 13–17, 2005, pp. 1060–1071.
[14] D.W. Scott, On optimal and data-based histograms, Biometrika 66 (3) (1979) 605–610.
[15] J. Kilpi, P. Lassila, L. Muscariello, On the dependence of Internet flow traffic, in: Proceedings of Second EuroNGI Workshop on New Trends in Modelling, Quantitative Methods and Measurements, 2005.
[16] <http://www.cs.waikato.ac.nz/ml/weka/>, 2008.
[17] <http://weka.sourceforge.net/manuals/weka.bn.pdf>, 2008.
[18] R. Beverly, A robust classifier for passive TCP/IP fingerprinting, in: Proceedings of PAM 2004, Juan-les-Pins, France, April 2004, pp. 158–167.
[19] G. Taleck, Ambiguity resolution via passive OS fingerprinting, in: Proceedings of Recent Advances in Intrusion Detection RAID 2003, 2003, pp. 192–206.
[20] J.C. Bolot, Characterizing end-to-end packet delay and loss in the internet, Journal of High-Speed Networks 2 (3) (1993) 305–323.
[21] V. Paxson, End-to-end internet packet dynamics, IEEE/ACM Transaction on Networking (TON) 7 (3) (1999) 277–292.
[22] W. Jiangm, H. Schulzrinne, Modeling of packet loss and delay and their effect on real-time multimedia service quality, in: Proceedings of NOSSDAV, June 2000.
[23] S.B. Moon, J. Kurose, D. Towsley, Correlation of packet delay and loss in the Internet, Technical Report 98-11, Department of Computer Science, University of Massachusetts, Amherst, MA, March 1998.
[24] G. Iannello, F. Palmieri, A. Pescapè, P. Salvo Rossi, End-to-end packet-channel bayesian model applied to heterogeneous wireless networks, in: Proceedings of IEEE GLOBECOM 2005, 2005, pp. 484–489.
[25] C. Demichelis, P. Chimento, IP packet delay variation metric for IP performance metrics (IPPM), RFC 3393, 2002.
[26] H. Oouch, T. Takenaga, H. Sugawara, M. Masugi, Study on appropriate voice data length of IP packets for VoIP network adjustment, in: Proceedings of IEEE GLOBECOM 2002, vol. 2, 2002, pp. 1618–1622.
[27] M. Claypool, J. Tanner, The effects of jitter on the perceptual quality of video, ACM Multimedia (2) (1999) 115–118.
[28] E.H.-K. Wu, M.-Z. Chen, JTCP: jitter-based TCP for heterogeneous wireless networks, IEEE Journal of Selected Areas on Communications 22 (4) (2004) 757–766.
[29] N. Duffield, F.L. Presti, V. Paxson, D. Towsley, Inferring link loss using striped unicast probes, in: Proceedings of IEEE INFOCOM, Anchorage, AK, April 2, 2001, pp. 915–923.
[30] N. Spring, R. Mahajan, T. Anderson, Quantifying the causes of path inflation, in: Proceedings of ACM SIGCOMM, Karlsruhe, Germany, 2003, pp. 113–124.
[31] N. Spring, R. Mahajan, D. Wetherall, Measuring ISP topologies with rocketfuel, IEEE/ACM Trasnaction on Networking (TON) 2 (1) (2004).
[32] M. Caesar, L. Subramanian, R. Katz, Towards localizing root causes of BGP dynamics, Technical Report UCB/CSD-04-1302, U.C. Berkeley, 2003.
[33] A. Feldmann, O. Maennel, Z.M. Mao, A. Berger, B. Maggs, Locating Internet routing instabilities, in: Proceedings of ACM SIGCOMM, Portland, OR, 2004, pp. 205–218.
[34] A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, H. Zhang, A clean slate 4D approach to network control and management, ACM SIGCOMM Computer Communication Review 35 (5) (2005) 41–54.
[35] Z. Ma, H.R. Shao, C. Shen, A new multi-path selection scheme for video streaming on overlay networks, in: Proceedings of IEEE International Conference on Communication, Paris, France, vol. 3, 2004, pp. 1330–1334.
[36] Alessio Botta, Antonio Pescapé, Vinh Q Bui, Weiping Zhu, An MDP-based approach for multipath data transmission over wireless networks, in: Proceedings of 2008 IEEE International Conference on Communications (ICC 2008), 2008.
[37] B. Krishnamurthy, H. Madhyastha, O. Spatscheck, ATMEN: A triggered network measurement infrastructure, in: Proceedings of fourteenth International World Wide Web Conference, Chiba, Japan, May 2005, pp. 499–509.
[38] R. Mortier, R. Isaacs, P. Barham, Anemone: using end-systems as a rich network management platform, Technical Report MSR-TR-2005-62, Microsoft Research, June 2005.
[39] V. Padmanabhan, S. Ramabhadran, J. Padhye, Client-based characterization and analysis of end-to-end internet faults, Technical MSR-TR-2005-29, Microsoft Research, Redmond, WA, March 2005.
[40] V. Padmanabhan, S. Ramabhadran, J. Padhye, NetProfiler: profiling wide-area networks using peer cooperation, in: Proceedings of Fourth International Workshop on Peer-to-Peer Systems, Ithaca, NY, March 2005.
[41] B. Tierney, J. Lee, B. Crowley, M. Holding, J. Hylton, F. Drake, A network-aware distributed storage cache for data intensive environments, in: Proceedings of IEEE High Performance Distributed Computing conference (HPDC-8), 1999, pp. 185–193.
[42] B. Noble, System support for mobile adaptive applications, IEEE Personal Communications Magazine 7 (2000) 44–49.
[43] Z.M. Mao, H.W. So, B. Kang, R.H. Katz, Network support for mobile multimedia using a selfadaptive distributed proxy, in: Proceedings of International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV-2001), 2001, pp. 107–116.
[44] Dapeng Wu, Y.T. Hou, Ya-Qin Zhang, Scalable video coding and transport over broadband wireless networks, Proceedings of the IEEE 89 (1) (2001) 6–20.
[45] Cisco Optimized Edge Routing (OER). <http://www.cisco.com/en/US/products/ps6628/products_ios_protocol_option_home.html., 2008.
[46] RadWare. LinkProof: A Traffic Manager for Multi-Homed Networks. <http://www.radware.com/content/products/lp/default.asp>, 2008.
[47] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris, Resilient overlay networks, ACM SIGOPS Operating Systems Review 35 (5) (2001) 131–145.
[48] Z. Duan, Z.L. Zhang, Y.T. Hou, Service overlay networks: SLAs, QoS, and bandwidth provisioning, IEEE/ACM Transaction on Networking (TON) 11 (6) (2003) 870–883.
[49] Z. Li, P. Mohapatra, Qron: Qos-aware routing in overlay networks, IEEE Journal of Selected Areas on Communications 22 (1) (2004) 29–40.
[50] L. Subramanian, I. Stoica, H. Balakrishnan, R. Katz, Overqos: an overlay based architecture for enhancing internet QoS, in: Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI), San Francisco, CA, USA, March 2004, pp. 71–84.
[51] X. Yang, NIRA: a new Internet routing architecture, ACM SIGCOMM Computer Communications Review 33 (4) (2003) 301–312.
[52] D. Katabi, I. Bazzi, X. Yang, A passive approach for detecting shared bottlenecks, in; Proceedings of IEEE International Conference on Computer Communications and Networks, 2001, pp. 174–181.
[53] D. Rubenstein, J. Kurose, D. Towsley, Detecting shared congestion of flows via end-to-end measurement, in: Proceedings of ACM SIGMETRICS, 2000, pp. 145–155.
[54] V. Padmanabhan, L. Qiu, H. Wang, Server-based inference of internet link lossiness, in: Proceedings of IEEE INFOCOM, vol. 1, 2003, pp. 145–155.
[55] S. Katti, D. Katabi, C. Blake, E. Kohler, J. Strauss, Multiq: automated detection of multiple bottleneck capacities along a path, in: Proceedings of ACM SIGCOMM Internet Measurement Conference, 2004, pp. 245–250.
[56] A.B. Downey, Using pathchar to estimate Internet link characteristics, in: Proceedings of ACM SIGMETRICS Conference on the Measurement and Modeling of Computer Systems, 1999, pp. 222–223.
[57] W. Wei, S. Jaiswal, J. Kurose, D. Towsley, Identifying 802.11 traffic from passive measurements using iterative bayesian inference, in: Proceedings of INFOCOM 2006, April 2006, pp. 1–12.
[58] R. Karrer, I. Matyasovszki, A. Botta, A. Pescapé, MagNets: experiences from deploying a joint research-operational next-generation wireless access network testbed, in: Proceedings of 3rd International Conference on Testbeds and Research Infrastructures (TridentCom), Orlando, FL, May 2007.
[59] <http://www.one-lab.org>, 2008.
[60] A.P. Dempster, A generalization of Bayesian inference, Journal of the Royal Statistical Society Series B 30 (1968) 205–247.
[61] Y.S. Huang, CY. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Transaction of Pattern Analysis and Machine Intelligence (1995).

Alessio Botta is a Ph.D. student in Computer Engineering and Systems at the Computer Science Department of University of Napoli "Federico II" (Italy), where he received the M.S. Laurea Degree in Telecommunications Engineering in 2004. His research interests fall in the areas of network measurements, traffic analysis, and network management with particular focus on performance evaluation and statistical characterization of wireless systems. He is a member of the IEEE.

Antonio Pescapé is Assistant Professor at the Department of Computer Engineering and Systems of the University of Napoli Federico II. He received the M.S. Laurea Degree in Computer Engineering and the PhD in Computer Engineering and Systems at University of Napoli Federico II. His research interests are in the networking field with focus on models and algorithms for Internet Traffic, Network Measurement and Management of heterogeneous IP networks, and Network Security. He has co-authored a large number of journal and conference publications. He is IEEE member and he has served and serves on several conference technical program committees (IEEE Globecom, IEEE ICC, IEEE WCNC, IEEE HPSR, etc.) and has served as Guest Editor of the Special Issue of Computer Networks on "Traffic classification and its applications to modern networks".

Giorgio Ventre is Professor of Computer Networks in the Department of Computer Engineering and Systems of the University of Napoli Federico II where he is leader of the COMICS team. COMICS stands for Computers for Interaction and Communications and is a research initiative in the areas of networking and multimedia communications. After started ITEM, the first research laboratory of the Italian University Consortium for Informatics (CINI), Giorgio Ventre is now President and CEO of CRIAI, a research company active in the areas of Information Technologies. As leader of the networking research group at University of Napoli Federico II Giorgio Ventre is principal investigator for several national and international research projects. His research interests are in the area of network protocols and architectures. Giorgio Ventre has co-authored more than 150 publications and he is member of the IEEE and of the ACM.