Identification of Network Bricks in Heterogeneous Scenarios

Alessio Botta, Antonio Pescapé, and Giorgio Ventre
Dipartimento di Informatica e Sistemistica, University of Napoli "Federico II"
{a.botta, pescape, giorgio}@unina.it

Abstract— **Accurate identification of network elements (*network bricks*) composing end-to-end paths represents a novel and interesting research topic. In heterogeneous scenarios, automatic *network bricks* identification can improve the performance of adaptive and network-aware applications. This work proposes an approach, based on *Bayesian* classifiers, for the identification of *network bricks* belonging to a large number of real heterogeneous end-to-end paths. The identification is performed by means of measurement and off-line observation of delay, jitter, and packet loss. We introduce the term "*blind* identification" meaning the capability to identify *network bricks*, by looking at Quality of Service (QoS) parameters observed on the end-to-end path. We propose first insights and preliminary results regarding the identification stage, based on both *concise* and *detailed* QoS parameters statistics. Moreover, we show some results of the identification performed using a reduced set of QoS parameters. Data traces and tools used in this work are publicly and freely available at `http://www.grid.unina.it/Traffic/`.**

I. Introduction

In current heterogeneous networks, end-to-end paths are composed of a wide range of different elements. Indeed, End user Devices (EuD), Operating Systems (OS), Access Networks (AN), and applications/protocols, exacerbate the heterogeneity of each path. Therefore, simple, efficient, and accurate methods for the identification of parts of the whole end-to-end path are useful to properly monitor, control, and manage networks. The automated knowledge of the above mentioned elements helps in efficiently tuning the adaptive application parameters over heterogeneous and mobile scenarios. For example, in a general end-to-end communication scenario, the knowledge of the access network at receiver side (eg, ADSL or UMTS), as well as the possibility to estimate the bitrate, helps in setting up efficient control and management activities (eg, peer selection in peer-to-peer applications). Acquiring this information in a direct way (i.e. asking to the application) is often not possible. Indeed, there are several situations in which the application can report erroneous information (eg, no direct knowledge, user trying to obtain different consideration, ...).

In this work we propose a framework for the *blind* identification of *network bricks* (see Section II for the definition of *network bricks*). With the term *blind*, we refer to an identification performed using parameters different from the network properties we are identifying. More precisely, our identification process is based on measures (actively) collected at the edge of each considered heterogeneous path. It is straightforward that the availability of accurate end-to-end measures is of great importance for the identification process.

In general, inferring network proprieties from end-to-end measurements represents an important and challenging task. In this framework, some interesting works have been proposed in literature. In [1], the authors propose a passive approach to detect bottlenecks in network paths. [2] presents results on detecting shared congestion of flows by means of end-to-end measurements. The authors of [3] propose inference techniques to estimate the loss rates of network links. The techniques are based on measures collected on a server. In [4] and [5], 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 similar to that presented in [6]. In this work, an iterative *Bayesian* technique, based on passive measurements, is used to identify 802.11 traffic. Differently from [6], we use an active approach to collect our measures. Such an approach has been used also in [7]. In this work, the authors classify the access networks in three types and show how it is possible to recognize the class elements using the outcomes of an active probing tool. Finally, several works use TCP/IP fingerprinting to detect host's characteristics. As an example, [8] uses a Bayesian classifier to passively detect host's operating system.

We would like to underline that the aim of this paper is not to present a tool for *network bricks* identification. The development of a such tool requires, in fact, several aspect to be considered (eg, intrusiveness of the measurement process, scalability, ...). We would rather assess the feasibility of identifying end-to-end path components by looking at Quality of Service (QoS) parameter statistics. More precisely, our approach aims at providing some experimental basis to demonstrate that a *blind* identification of different "*network bricks*" is possible. To support our thesis we present some results of the identification framework we set up. In particular, first we present *network bricks* identification results (over a number of end-to-end heterogeneous paths). Then, taking into account the relationships among QoS parameters, we present identification results over a reduced set of QoS parameters.

The rest of the paper is organized as follows. Section II provides some definitions to clarify our approach, and gives details about the paper contribution. Section III contains a short description of the analytical tools we used in the identification stage. In Section IV we give an overview of considered end-to-end paths, measurement methodology, used data traces, and statistical tools adopted to determine the discriminators. Section V contains preliminary results on *network bricks*

1-4244-0419-3/06/$20.00 ©2006 IEEE

TABLE I
NETWORK BRICKS.

	Protocol	Sender OS	Receiver OS	Sender AN	Receiver AN	Bitrate	Sender EuD	Receiver EuD
1	UDP	Linux	Linux	Ethernet	Ethernet	51,2 kbps	Workstation	Workstation
2	TCP	Windows	Windows	WLAN 802.11b	WLAN 802.11b	409,6 kbps	PC Desktop	PC Desktop
3	SCTP	Linux Familiar	Linux Familiar	GPRS	GPRS	819,2 kbps	Laptop	Laptop
4				UMTS	UMTS		Palmtop	Palmtop
5				ADSL	ADSL			

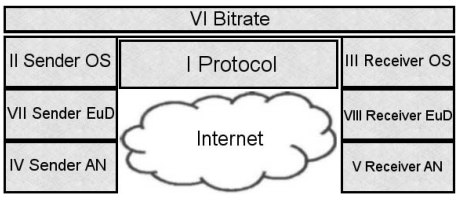

Fig. 1. Abstraction of Network Scenarios and *Network Bricks*.

identification. Also, it presents results after a reduction of the number of the considered QoS parameters. Finally, Section VI ends the paper with conclusions and issues for future research.

II. PROBLEM DEFINITION AND APPROACH

A *blind* identification, based on QoS metrics, of network bricks permits to identify network elements without physically dealing with them (eg, identify elements that are physically unreachable). In general, active or passive measurements can be used to accomplish this task.

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

DEFINITION 1: *Network brick*. With the term *network brick* we mean a network element (or a device component) from those shown in Table I and graphically represented in Figure 1.

DEFINITION 2: *End-to-end path*. With the term *end-to-end path* we mean a network path composed of *network bricks* (see Figure 1). Differently from the literature, our definition of end-to-end path includes also elements like bitrate of the path, sender and receiver end user devices, their operating systems, and, finally the used protocol (see Section IV-A).

DEFINITION 3: *QoS parameter*. With the term QoS parameter we mean a parameter among delay, jitter, and packet loss. Each of them is used to calculate the adopted discriminators (see Section IV-B).

DEFINITION 4: *Discriminators*. With the term discriminators we mean a set of statistical values calculated over each QoS parameter data trace. In this work we consider the discriminators reported in Table III.

Using the previous definitions, we propose a framework where, by looking at the statistics of QoS parameters, it is possible to identify *network bricks* of a number of heterogeneous end-to-end paths. The identification process uses as discriminators a set of QoS parameter statistics we divided in *concise* and *detailed statistics* (see Table III).

Even if the *network bricks* identification constitutes our principal contribution, our work consists of three main parts carried out in three successive phases. First, we collect several traces of some selected QoS parameters over a broad range of heterogeneous end-to-end paths. The traces are collected with an active measurement approach over the heterogeneous network depicted in Figure 2. More details regarding the adopted measurement approach are provided in Section IV-B. Second, after data acquisition and sanitization, we calculate some statistics for each considered QoS parameter. The statistics are selected looking at their capability to identify *network bricks*. Third, using these statistics in a supervised classification algorithm, we identify the *network bricks* composing the end-to-end paths. Moreover, taking into account the relationships between the considered QoS parameters extensively discussed in literature, we show how it is possible to discard one of them without affecting the overall identification accuracy (i.e. the percentage of *network bricks* correctly identified). This result appears extremely interesting when just two of the three parameters are accessible or measurable (eg, in network scenarios where there is loose or no synchronization between sender and receiver the one way delay is unmeasurable).

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 considered QoS parameter *detailed statistics* represent a complete and robust set of discriminators; (III) we show that the reduction of the set of discriminators slightly affects the overall accuracy; (IV) we present a proof of concept over a number of real end-to-end heterogeneous paths; (V) we make the data traces we collected freely available at [9].

III. ANALYTICAL BASIS: A BRIEF OVERVIEW

To achieve its goal, the identification process requires a supervised classification algorithm. Despite this, 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 use Bayesian classifiers. In particular, we use *Bayesian Network* (BN) classifiers and, to check a first proof of result generalization, we also consider *Naïve Bayesian* (NB) classifiers. The motivations at the base of this choice are their simplicity, spread diffusion, and their ability to work in an intuitively fashion. Additionally, we choose the simplest of the computational methods in order to ensure an acceptable processing time [10]. To introduce the classifiers we used, in this Section, we provide a brief overview of *Bayesian Network* and *Naïve Bayesian* classifiers.

A. Bayesian Network Classifier

Let $S = \{x_1, ..., x_n\}$ with $n \geq 1$ be a set of variables. A *Bayesian Network* (BN) over the set S is a directed

TABLE II

COORDINATES OF CONSIDERED END-TO-END PATHS.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
I	Protocol	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
II	Sender OS	1	1	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	2	2	2	1	1	1	2	2	2	1
III	Receiver OS	1	1	2	2	1	1	2	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	1
IV	Sender AN	5	5	3	3	3	3	4	4	4	4	4	4	4	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2
V	Receiver AN	1	1	1	1	2	2	3	3	3	1	1	2	2	5	5	3	3	3	3	3	3	3	3	3	3	3	3	2
VI	Bitrate	2	1	3	2	3	2	3	2	1	2	1	2	1	3	2	3	2	1	3	2	1	3	2	1	3	2	1	1
VII	Sender EuD	2	2	3	3	3	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3
VIII	Receiver EuD	1	1	1	1	3	3	3	3	3	1	1	3	3	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3

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 *Bayesian Network* represents the chain rule for a joint probability distribution

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

In general, the classification consists in assigning a set of variables $x = x_1, ..., x_n$, called *attribute variables*, to some other variables $y = x_0$, called the *class variable*. The classifier $b : x \rightarrow y$ 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). Therefore, the learning stage is very important. For this classifier, it consists in finding the appropriate *Bayesian Network* 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 *Bayesian Network*. Observing that

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

and that all variables in x are known, no complicated inference algorithms are necessary. It is sufficient to calculate Equation (1) for all class values. For further details refer to [11].

B. Naïve Bayesian Classifier

Let $x = \{x_1, ..., x_n\}$ be a data sample representing a realization of $X = \{X_1, ..., X_n\}$, and let each random variable X_i be described by m attributes $\{A_1, ..., A_m\}$, then $X_i = (A_1^{(i)}, ..., A_m^{(i)})^T$ is a random vector. Let $C = \{c_1, ..., c_h\}$ be the set of classes of interest. For each observation x_i in x, there is a mapping C : $x \rightarrow C$ 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 rules provides

$$p(c_j|y) = \frac{p(c_j) \cdot f(y|c_j)}{\sum_{c_j} p(c_j) \cdot f(y|c_j)} \quad (3)$$

where $p(c_j)$ represents the *a priory probability* of class c_j, $f(y|c_j)$ represents the conditional (given c_j) probability of y, 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 $f(y|c_j), j = 1, ..., h$ given some training set x. To achieve this goal, Naïve Bayes makes some assumptions on $f(\cdot|c_j)$ such as the independence of A_i's and the standard Gaussian behavior of them. The problem is then reduced to estimating the parameters of the Gaussian distribution and $p(c_j)$. Despite its simplicity, *Naïve Bayes* has been shown to work better than more complex methods and to be able to cope with complex situations [12].

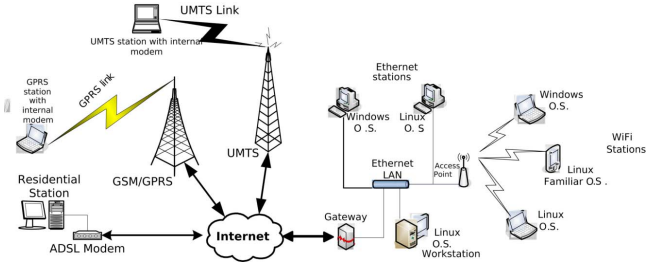

Fig. 2. Network Test-bed.

IV. PATHS, TOOLS, METRICS, AND DISCRIMINATORS

As already said in Section II, our work is carried out in three main steps.

First, we conduct an active measurement stage on the heterogeneous network depicted in Figure 2. In Section IV-A, the testbed we used and the considered end-to-end paths are described. This measurement stage allows us to collect a number of data traces related to three QoS parameters (that are jitter, delay, and packet loss). In Section IV-B details about measurement methodology and data traces are given.

Second, we select 13 statistics (dividing them in *concise* and *detailed*) that are able to capture the peculiar characteristics of the *network bricks* we identify. We then calculate these statistics using all the data traces collected in the previous stage (in an off-line fashion). In this way, we obtain the discriminators we use in the successive step. Section IV-C contains a description of considered discriminators and the motivations for which we chose them.

Third, using the discriminators previously calculated, we adopt supervised classification algorithms to identify the network bricks. Details regarding the used tool and the way we perform the identification task are provided in Section IV-D.

A. Network Scenario and End-to-end paths

Figure 2 shows the network test-bed on which we applied our measurement and identification approach. The test-bed

TABLE III
USED DISCRIMINATORS.

Concise statistics						Detailed statistics						
F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**F12**	**F13**
Mean	Median	Standard Deviation	Min	Max	IQR	r(2)	r(10)	$\tau(2)$	$\tau(10)$	$s(2)$	$s(10)$	Entropy

comprises different heterogeneous wireless/wired networks, as well as different Devices and Operating Systems. The details of the hardware we used are reported in Table IV. The measurements are carried out considering several possible combinations of these variables. In particular, we perform throughput, jitter, delay, and packet loss measurements by varying all the possible testbed parameters (that are Operating System, End User Device, Access Network, Transport Protocol, and Traffic Condition).

TABLE IV
DEVICES DESCRIPTION.

Devices	Description
Laptop 1	Toshiba Satellite Pro 4300, Intel PIII 650 Mhz, 186MB RAM, 256KB Cache, Windows 2000 Prof. O.S.
Laptop 2	Toshiba Satellite S5200-801, Intel P4 2.0 Ghz, 512MB RAM, 512KB Cache, Windows XP Prof. O.S.
PC 1	Intel P4 2.6 Ghz, 1024MB RAM, 512KB Cache, Windows XP Prof. O.S.
GPRS Modem	Merlin G301 - Novatel Wireless
UMTS Modem	Merlin U530 - Fast Mobile Card 3
ADSL Modem	D-Link DSL-502T
WLAN NIC	D-Link Air-Plus
Ethernet NIC	3Com EtherLink XL 10/100

For the purpose of *bricks* identification, the selected network scenario exhibits a wide parameter space composed of a large number of variables (see also Table I) to setup, configure, and analyze. Mixing all variables reported in Table I 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 28 end-to-end paths (with 20 observations for each path).

The details of the considered paths are reported in Table II. In such a Table, 8 coordinates for each path are contained. Their meaning is shown in the columns of Table I and in Figure 1. More precisely, each column of Table II represents the coordinates of a path we consider. The meaning of these coordinates is contained in Table I. As an example, consider the column of Table II named 1. This column represents the coordinates of the first path we consider, that are $(1, 1, 1, 5, 1, 2, 2, 1)$. Looking at Figure 1, we can see that the first coordinate represents the protocol used in the end-to-end path. According to Table I, the value of 1 means that the protocol is UDP. Following this approach, the Sender OS (second coordinate) is Linux. The receiving OS (third coordinate) is Linux, and so on.

From all the possible end-to-end paths, in this work we consider just those using UDP and not containing Palmtop (as end user device) or Linux Familiar[1] (as Operating System). This allows us to (i) restrict our attention to a reduced set of paths and to (ii) include packet loss statistics in our set of discriminators.

[1]Open Source Operating System for PDA devices

B. Measurements Approach and Data traces

Data traces of QoS parameters we used in this work are collected by means of an active measurement approach using D-ITG [13]. It is a synthetic traffic generator able to produce a number of traffic patterns by modeling PS (*Packet Size*) and IDT (*Inter Departure Time*) random processes. Therefore, it is capable of generating synthetic traffic that is, at the same time, realistic. In order to reduce the number of variables to consider, we use only Constant Bitrate (CBR) traffic generated with constant PS and IDT. The measurements are performed by using three traffic conditions named *Low*, *Medium*, and *High Traffic* respectively. These traffic conditions differ, from each other, in terms of the used IDT, that are $1/100$ s, $1/1000$ s, and $1/10000$ s respectively. For each IDT different PS, ranging from 64 to 1500 Bytes, are used.

Due to the nominal bandwidth of some AN (GPRS and UMTS), for the *network bricks* identification we consider only *Low Traffic* condition (IDT = $1/100$ s) with PS ranging in $\{64, 512, 1024\}$ Bytes. With PS of 64 Bytes the wireless (GPRS and UMTS) channel is far from saturation. Instead, with a PS equal to 1024 Bytes it is in saturation.

Traffic is generated for a duration of 120 seconds. This duration is chosen to correctly evaluate performance out of the transient phase. Each log file has a size of about 1.5 MByes. The measurement stage has been performed in the time period between December 2003 and November 2004, in the day hours between 9:00 am and 6:00 pm. To avoid measure polarization due to external causes of errors, the measurements have been opportunely interleaved. In the measurement stage, over 34 GB of traffic traces have been collected. Such traces have been previously inspected and sanitized in order to detect and remove samples affected by errors. Also, it is worth noting that the used GPRS and UMTS connections have been provided by two of the principal Italian Telecom Operators. Such connections are the same provided to all their customers; for this reason, the used data is related to that a common user would experience.

1) ***Data Archives****:* At [9] we made freely available several archives containing outcomes of measurements over real networks. Each archive contains files with samples of QoS parameters measured over several end-to-end paths. Samples are obtained, by adopting the above described active measurement approach, sending probe packets with a rate of 100 pps and size ranging in $\{64, 512, 1024\}$ Bytes. More details about the traffic parameters are contained in Table V.

Each sample is calculated using non-overlapping windows of 10 ms length. At [9] we provide also archives containing samples calculated on a per packet basis.

TABLE V
TRAFFIC PARAMETERS.

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

C. Discriminators

Using the samples contained in each QoS parameter trace, we calculated 13 discriminators (see Table III) in an off-line fashion. They represent a small set of statistics, we divided in *concise* and *detailed* statistics, able to correctly identify the characteristics of the considered "*network bricks*". *Detailed statistics* permit to better understand the behavior of QoS parameters [14] [15].

As for the *concise statistics*, we consider well know parameters like the minimum, maximum, mean, 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, the IQR and median value are better estimators for skewed distributions, than the standard deviation and the average value respectively. They are less influenced by extreme samples.

As for the *detailed statistics*, we consider the entropy and three types of correlation coefficients. Generally speaking, entropy is a measure of the uncertainty of a random variable X. It is defined as

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

It was used to classify network links in [7] too. As regards correlation coefficients, we use three correlation measures: *Pearson*, *Spearman*, and *Kendall* correlation. The most spread 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 the two random variable X and Y. It ranges from -1 to +1. The correlation of Spearman (s) measures the linear relation existing between two variables. It differs from the previously analyzed correlation of Pearson only in that the calculations are done after changing the numbers into ranks. Spearman's correlation can be therefore evaluated by means of Equation (5) using the ranked data.

Linear correlations have some disadvantages. In particular, they are sensitive to outliers and they measure the "average dependencies" of random variables. To overcome these limitations and to properly take into account upper tail dependencies between variables, we also consider Kendall's correlation. Let $(\widetilde{X}, \widetilde{Y})$ be an independent copy of (X, Y). Two pairs (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 Equation (6) that can be estimated as in Equation (7).

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

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

Thanks to its properties, Kendall's tau has been already used in the study of traffic flows dependence in [16]. In this work all the above cited correlation coefficients are evaluated on samples of the same variable (auto-correlation) at both lag 2 and lag 10. This permits to evaluate the extent to which the samples of the considered variables are dependent.

The simple set of discriminators we just presented provide us the possibility to identify the *network bricks* of a large number of end-to-end heterogeneous paths.

D. Identification tool

We provide identification results using version 3.4.7 of WEKA [17], an intuitive and complete software for solving classification and clustering problems. We use the 50% *percentage split* option of WEKA software for our identification stage. For each path, 20 instances are considered. Therefore, 10 of them are used in the classifier training (or learning) stage, while the other 10 instances are used in the testing stage.

The identification process can be summarized as follows. For each *network brick* we identify, we first instruct the classification algorithm. This learning stage is performed using 280 (10 realizations for each of the 28 paths) vectors of discriminators (instances). In a successive stage, classification phase, we use another 280 instances (test set) for testing the identification process. These two sets are called learning set and test set respectively. The former, learning set, represents the attributes (see Section III) the classifier uses to build its model. In such phase, the classifier discovers the peculiar characteristics, in terms of discriminator values, of each class. Using those characteristics, it attributes, in the classification stage, the elements of the test set to some class. To perform the identification, we instruct the classifier to consider each brick instance (eg, UDP for 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 ascribes it to the correct class. For each network brick, using the computer named PC1 in Table IV, the identification stage takes about 1.5 seconds.

V. EXPERIMENTAL RESULTS

A. Blind identification

Table VI contains our preliminary results. For each *network brick*, we report the percentage of correctly identified instances. Moreover, in this Table the percentage of identified instances is presented for the two considered classifiers. Also, for each *brick* and for each classifier, two values are reported.

The first one is related to an identification based on a smaller set of discriminators (*concise statistics*), while the second one is obtained by using a larger set of discriminators that includes also the detailed parameters (*concise* plus *detailed statistics*). The components of these two sets are indicated in Table III.

As shown in Table VI, the identification proves to be more accurate when performed by using the complete set of discriminators. For this reason, in the following we discuss just the results related to this kind of identification. Moreover, for the sake of brevity, comments are given just for the *Bayesian Network* results. Similar consideration can be done for the *Naïve Bayes* results.

As one would expect, some 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 *Receiver OS*. In this case, the percentage of correctly identified instances is equal to 93%. The *Receiver AN*, *Receiver EuD*, and *Bitrate* achieve the same results in terms of identification accuracy. For all of them, the percentage of correctly identified instances is indeed equal to 86%. The percentage of *Sender AN* instances correctly identified is equal to 82%. The *Sender OS* identification presents the 79% of correctly identified instances. Finally, 75% of *Sender EuD* instances are correctly assigned to the related class.

Obtained results show that the blind identification of *network bricks* is possible and its accuracy is dependent on the considered *brick*. It ranges from 75% to 93% therefore providing satisfying results. The motivations at the base of such variation are different. First, if we compare the results obtained for the *network bricks* related to sender and receiver hosts (*Operating System*, *Access Network*, and *End User Device*) we can see that in the first case the accuracy is lower than in the second one. This behavior is probably due to the fact that, because our data is collected at the receiver side of the communication, the collected statistics are more influenced by the receiver host. Moreover, we can observe that the OS is one of the *bricks* whose identification is more accurate. If we look at the values reported in Table II (II and III coordinates), we can see that the considered end-to-end paths comprise Windows and Linux OSs. The different way these two OSs manage network related operations (TCP/IP stack implementation, queue management ...) is at the base of this result. Finally, the bitrate identification is highly accurate. This is an interesting result. It witnesses that the bitrate influences several parameters of the end-to-end communication making it possible its identification by looking at delay, jitter, and packet loss. Finally, the different percentages achieved by the *bricks* are probably due also to the nature of our traces. Despite this, the overall result witnesses that the *network bricks* identification reveals promising.

B. Blind Identification with QoS parameters reduction

A number of works, studying the dependencies among the considered QoS parameters, are present in literature. According to our experience in managing networks, and taking into account these works, we propose a methodology aiming to withdraw one the three QoS parameters considered in Section V-A. First, we discuss results of a blind identification with a reduced set of QoS parameters. Second, in Section V-B.1 we justify our intuition, quantifying the dependences between QoS parameters (using information theory based tools).

The relationships between different QoS parameters have been extensively studied in literature. As for packet loss and delay, a number of works have shown the dependencies between them [18], [19], [20], [21], [22]. In [21] the authors analyzed the correlation between packet loss and network delay considering that those events occur in sequence. In [22] a model, based on Hidden Markov Models, able to jointly model packet loss and delay over heterogeneous network paths, is presented. Several studies investigate the relationships between packet loss and jitter. In [23] the authors found that a high traffic load can cause packet loss and jitter. In [24] Wu and Chen propose a jitter-based scheme to adapt sending rate to packet loss and jitter ratios. In [25] authors found that jitter degrades perceptual quality nearly as much as packet loss does.

Finally, a clear dependence exists between delay and jitter. In this work the jitter samples are calculated using the following formula

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

where j_k is the *k-th* jitter sample and d_k is the one way delay experimented by the *k-th* received packet. This formula is compliant with the definition given in [26].

Starting from the previous considerations, we perform a *network bricks* identification using the discriminators from two out of the three considered QoS parameters. More precisely, we apply our *blind* identification process using the following pairs: (i) packet loss and jitter; (ii) packet loss and delay; (iii) jitter and delay.

Table VII shows that the identification based on two QoS parameters is still possible. The overall identification accuracy is indeed almost preserved. This confirms that the QoS parameters are dependent. Table VII shows also that this dependency differently influences the different *bricks*. In particular we can observe that, for the *Sender* and *Receiver OS bricks*, the identification performs equally for the three QoS parameter pairs. This means that the identification algorithm gives the same importance to the three parameters. In the case of *Sender*/*Receiver AN* and *Sender*/*Receiver EuD*, the accuracy decreases mainly when the packet loss discriminators are not taken into account. For these *bricks* a minor decrease is noticed also when the jitter discriminators are discarded. The *Bitrate* identification accuracy seems to be more dependent on the delay discriminators. As in the previous case, the accuracy is slightly influenced by the jitter too.

*1) **Quantifying dependencies between QoS parameters**:* In the previous Section (V-B), we have supposed that some relationship exists between QoS parameters. According to this hypothesis, we have withdrawn one of the three considered QoS parameters, showing that the accuracy of the identification process remains almost unchanged. In this Section

TABLE VI

IDENTIFICATION RESULTS: PERCENTAGE OF IDENTIFIED INSTANCES.

	Bayesian Network (BN)		Naïve Bayes (NB)	
Brick	Concise Parameters	Concise + Detailed Parameters	Concise Parameters	Concise + Detailed Parameters
Sender OS	79%	79%	64%	82%
Receiver OS	93%	93%	64%	79%
Sender AN	79%	82%	68%	86%
Receiver AN	75%	86%	68%	79%
Bitrate	57%	86%	64%	89%
Sender EuD	75%	75%	57%	82%
Receiver EuD	86%	86%	75%	82%

TABLE VII

RESULTS OF THE IDENTIFICATION BASED ON TWO QOS PARAMETERS.

	packet loss/jitter		packet loss/delay		delay/jitter	
Brick	BN	NB	BN	NB	BN	NB
Sender OS	71%	79%	71%	79%	79%	79%
Receiver OS	79%	93%	71%	93%	82%	93%
Sender AN	79%	79%	75%	79%	79%	79%
Receiver AN	79%	86%	71%	82%	82%	75%
Bitrate	89%	79%	89%	86%	89%	82%
Sender EuD	71%	75%	75%	75%	89%	64%
Receiver EuD	68%	86%	82%	82%	82%	86%

we provide an information theory-based validation of the previously supposed thesis, also useful to quantify the dependencies between parameters. In particular, the *Symmetrical Uncertainty* (a measure from the information theory) and the Correlation Coefficient (r as in Equation (5)) are used to achieve this goal.

The *Symmetrical Uncertainty* is based on the concept of *Entropy* (see Equation (4)) and the concept of *Information Gain*. The *Conditional Entropy* of X given Y is defined as

$$H(X|Y) = -\sum_{y \in Y} \sum_{x \in X} P(x|y) \cdot \log_2 P(x|y) \qquad (9)$$

The *Information Gain* is defined as

$$I_G(X|Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \qquad (10)$$

I_G represents the amount of information gained about X by observing Y. As shown in Equation (10), it is a symmetrical estimate, therefore it also represents the amount of information gained about Y by observing X. Finally, the *Symmetrical Uncertainty* (SU) is defined as

$$SU(X,Y) = 2 \cdot \frac{I_G(X|Y)}{H(X) + H(Y)} = SU(Y,X) \qquad (11)$$

It ranges from 0 to 1, where the value 0 means that X and Y are independent whereas the value 1 suggests that the knowledge of Y implies the knowledge of X. Despite the information gain, SU is not influenced by random variables with more samples. To obtain a correct SU estimate, it is necessary to evaluate the appropriate number of bins necessary to represent the probability density function (pdf). In this work, to find such an appropriate number, a preliminary analysis is conducted. Thanks to it, we learn that, as shown in Figure 3,

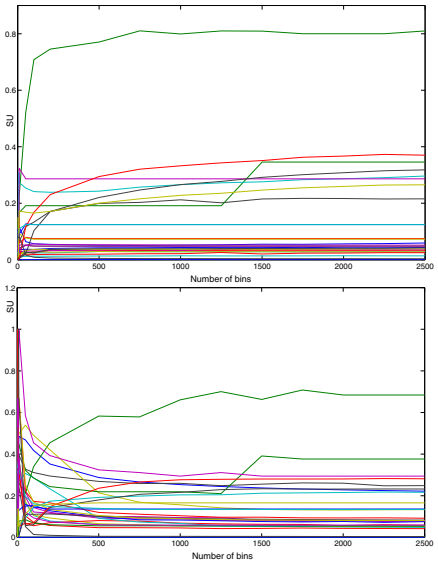

Fig. 3. Packet Loss/Delay (left) and Packet Loss/Jitter (right) SU trend.

the SU presents an increasing trend when sketched against the number of bins of the two variables (X and Y). More precisely, Figure 3 shows that, despite their differences, all the curves present a variable trend for small bin number values[2]. This trend finally stabilizes when a sufficiently high value is reached. This result implies that, the SU could be wrongly estimated if a such analysis is not taken into account.

Table VIII contains the average values of SU and Correlation Coefficient (r) calculated for each of the considered end-to-end paths. In this case, the SU is evaluated using 2000 bins. Moreover, for each path, we report the SU and r values calculated for all the pairs of QoS parameters. Values in Table VIII and the trends sketched in Figure 3 demonstrate that SU and r values are different for different paths. More interesting, this Table permits to identify the paths for which the identification process, based on two QoS parameters, produces wrong results. As an example, let us consider the path labeled P11. For this configuration the SU value calculated between packet loss and delay is the lowest. P11 is indeed the path which instances are wrongly classified when the identification is performed with no discriminators related to the delay (see Table VII).

[2]This figure witnesses the SU trend. The legend is not included because there is no need to identify single plots. For numerical details see Table VIII.

TABLE VIII

SYMMETRICAL UNCERTAINTY AND CORRELATION COEFFICIENT BETWEEN QOS PARAMETERS.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
Packet Loss and Delay SU	0.06	0.01	0.03	0.03	0.05	0.03	0.22	0.04	0.80	0.02	0.01	0.04	0.04	0.04
Packet Loss and Delay r	-0.23	0.01	-0.16	-0.14	-0.27	-0.16	0.11	0.11	0.36	-0.09	-0.04	-0.13	-0.03	-0.11
Packet Loss and Jitter SU	0.23	0.01	0.04	0.05	0.08	0.05	0.26	0.08	0.70	0.06	0.06	0.08	0.13	0.08
Packet Loss and Jitter r	0.12	-0.01	0.91	0.87	0.91	0.87	0.34	0.03	0.83	0.05	0.27	0.23	0.66	0.05
Delay and Jitter SU	0.45	0.37	0.82	0.70	0.76	0.68	0.91	0.68	0.83	0.53	0.28	0.60	0.29	0.32
Delay and Jitter r	-0.33	0.09	-0.09	-0.16	-0.29	-0.20	-0.09	-0.09	0.24	-0.24	-0.04	-0.28	0.03	0.12
	P15	**P16**	**P17**	**P18**	**P19**	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**	**P28**
Packet Loss and Delay SU	0.01	0.35	0.07	0.29	0.29	0.07	0.31	0.12	0.05	0.37	0.12	0.05	0.26	0.01
Packet Loss and Delay r	0.01	-0.09	-0.13	0.23	-0.16	-0.10	-0.07	-0.10	-0.02	0.14	-0.06	0.01	0.23	0.27
Packet Loss and Jitter SU	0.01	0.38	0.09	0.22	0.29	0.08	0.23	0.14	0.05	0.28	0.13	0.06	0.17	0.01
Packet Loss and Jitter r	-0.01	0.94	0.98	0.13	0.41	0.99	0.86	-0.19	0.06	0.83	0.74	0.88	0.11	0.94
Delay and Jitter SU	0.32	0.95	0.88	0.67	0.98	0.94	0.61	0.95	0.94	0.68	0.96	0.89	0.58	0.71
Delay and Jitter r	0.34	-0.18	-0.08	0.01	-0.30	-0.09	-0.13	-0.19	0.14	-0.01	-0.02	0.04	0.03	0.30

Comparing the different QoS parameter pairs, we can see that delay and jitter have achieved higher values of SU and r. This is consistent with their mathematical relation (see Equation (8)). Despite these higher values, the identification process based on delay and jitter achieves the same accuracy (in almost all cases) of the other two pairs.

VI. CONCLUSIONS AND ISSUES FOR RESEARCH

In this paper we studied the problem of (*blind*) identification of network elements (we called them *network bricks*) over heterogeneous wired/wireless networks. The proof of concept results shown in this paper confirm that our idea of *blind* identification of *network bricks* is feasible. Also, we demonstrated how a complete set of statistical discriminators perform better that a reduced set. Furthermore, we have shown that *blind* identification is possible also with a reduced set of QoS parameters. It represents a *per se* result, also useful when only a limited set of QoS parameters is available. To increase the number of path instances, our ongoing work deals with performing an increasing number of measurements over the considered end-to-end paths. This allows to confirm the preliminary identification results shown here. Finally, it is worth to underline that our current implementation of the identification process operates in an off-line fashion. It requires, in fact, the data traces to be previously collected and preprocessed. In this stage we were interested in testing the applicability and suitability of our idea of identification. At present, we have still to investigate the implications related to developing our approach in an on-line fashion. This change of modus operandi requires several aspects to be investigated (eg, intrusiveness of the measurement process, scalability, ...).

VII. ACKNOWLEDGEMENTS

This work has been partially supported by the MIUR in the framework of the PRIN 2004 Quasar Project, by the Content EU NoE, and by Netqos and OneLab EU projects.

REFERENCES

[1] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks", IEEE International Conference on Computer Communications and Networks, 2001, pp. 174 - 181.

[2] D.Rubenstein, J.Kurose, D.Towsley, "Detecting shared congestion of flows via end-to-end measurement", ACM SIGMETRICS, June 2000, p.145-155.

[3] V. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of internet link lossiness", IEEE INFOCOM 2003, Vol. 1, pp. 145 - 155.

[4] S. Katti, D. Katabi, C. Blake, E. Kohler, and J. Strauss, "Multiq: Automated detection of multiple bottleneck capacities along a path", ACM Sigcomm Internet Measurement Conference, 2004, pp.245-250.

[5] A. B. Downey, "Using pathchar to estimate Internet link characteristics", Measurement and Modeling of Computer Systems, pp. 222-223, 1999.

[6] W. Wei, S. Jaiswal, J. Kurose, D. Towsley, "Identifying 802.11 Traffic from Passive Measurements Using Iterative Bayesian Inference", accepted at IEEE INFOCOM 2006 (April).

[7] W.Wei, B.Wang, C.Zhang, J.Kurose, D.Towsley, "Classification of Access Network Types: Ethernet, Wireless LAN, ADSL, Cable Modem or Dialup?", IEEE INFOCOM 2005, Miami, March 13-17, pp. 1060-1071

[8] R. Beverly, A Robust Classifier for Passive TCP/IP Fingerprinting, PAM 2004, pp. 158-167, Juan-les-Pins, France, April 2004

[9] `http://www.grid.unina.it/Traffic/` [Online]

[10] A.W.Moore, D.Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques", ACM SIGMETRICS 2005, Banff, pp.50-60.

[11] Remco R. Bouckaert, "Bayesian networks in Weka". Technical Report 14/2004. Computer Science Department. University of Waikato. 2004.

[12] I.H. Witten, E. Frank, "Data Mining", Morgan Kaufmann Publishers, 2000.

[13] `http://www.grid.unina.it/software/ITG/` [Online]

[14] Q. Li, D. L. Mills, "On the long-range dependence of packet round-trip Delays in Internet", ICC 98, Vol. 2, pp. 1185-1191

[15] M.S. Borella, S. Uludag, G.B. Brewster, I. Sidhu, "Self-similarity of Internet packet delay", ICC 97 V. 1, June 97 pp. 513–517

[16] J. Kilpi, P. Lassila, L. Muscariello, "On the Dependence of Internet Flow Traffic", Second EuroNGI Workshop on New Trends in Modelling, Quantitative Methods and Measurements", November, 2005.

[17] `http://www.cs.waikato.ac.nz/ml/weka/` [Online]

[18] J.C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet", Journal of High-Speed Networks, V. 2(3), pp. 305-323, Dec. 1993.

[19] V. Paxson, "End-to-End Internet Packet Dynamics", IEEE Trans. on Networking, V. 7(3), pp. 277-292, June 1999.

[20] W. Jiang, H. Schulzrinne, "Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality", NOSSDAV, June 2000.

[21] S.B. Moon, J. Kurose, D. Towsley, "Correlation of Packet Delay and Loss in the Internet". Technical Report 98-11, Dep. of Computer Science, University of Massachusetts, Amherst, MA 01003.

[22] G. Iannello, F. Palmieri, A. Pescapè, P. Salvo Rossi, "End-to-End Packet-Channel Bayesian Model applied to Heterogeneous Wireless Networks", IEEE GLOBECOM 2005, pp. 484-489, Nov. 2005.

[23] H. Oouch, T. Takenaga, H. Sugawara, M. Masugi, "Study on appropriate voice data length of IP packets for VoIP network adjustment", IEEE GLOBECOM 2002. pp. 1618-1622 vol.2, Nov. 2002

[24] E. H.-K. Wu and M.-Z. Chen, "JTCP: Jitter-Based TCP for Heterogeneous Wireless Networks", IEEE JSAC, May 2004, Vol. 22, N. 4, pp. 757-766.

[25] M. Claypool, J. Tanner, "The effects of jitter on the perceptual quality of video", ACM Multimedia (2) 1999: pp. 115-118

[26] C. Demichelis, P. Chimento, IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), RFC 3393, November 2002