A Survey on Internet Traffic Identification

Arthur Callado, Carlos Kamienski *Member, IEEE,* Géza Szabó, Balázs Péter Gerő, Judith Kelner, Stênio Fernandes *Member, IEEE,* and Djamel Sadok, *Senior Member, IEEE*

***Abstract*—The area of Internet traffic measurement has advanced enormously over the last couple of years. This was mostly due to the increase in network access speeds, due to the appearance of bandwidth-hungry applications, due to the ISPs' increased interest in precise user traffic profile information and also a response to the enormous growth in the number of connected users. These changes greatly affected the work of Internet Service Providers and network administrators, which have to deal with increasing resource demands and abrupt traffic changes brought by new applications. This survey explains the main techniques and problems known in the field of IP traffic analysis and focuses on application detection. First, it separates traffic analysis into packet-based and flow-based categories and details the advantages and problems for each approach. Second, this work cites the techniques for traffic analysis accessible in the literature, along with the analysis performed by the authors. Relevant techniques include signature-matching, sampling and inference. Third, this work shows the trends in application classification analysis and presents important and recent references in the subject. Lastly, this survey draws the readers' interest to open research topics in the area of traffic analysis and application detection and makes some final remarks.**

***Index Terms*—Measurement, Application Identification, Traffic Analysis, Classification.**

I. Introduction

CHARACTERIZATION of Internet traffic has become over the past few years one of the major challenging issues in telecommunication networks [1]. It relies on an in-depth understanding of the composition and the dynamics of Internet traffic, which is essential in management and supervision of the ISP's (Internet Service Provider) network. Furthermore, the increased capacity and availability provided by broadband connections has led to a more complex behavior for a typical user, very different from a traditional dial-up user.

In general, the characterization of Internet traffic provides insights for various network management activities, such as capacity planning and provisioning, traffic engineering, fault diagnosis, application performance, anomaly detection and pricing. There have been some recent efforts on measuring and analyzing Internet traffic. Most of them pointed out that currently the predominant type of traffic is produced by peer-to-peer (P2P) file sharing applications, which can be responsible for more than 80% of the total traffic volume depending on the location and hour of day. In more recent analysis (yet unpublished) [2], last year's Video sharing traffic has greatly increased Internet usage, surpassing P2P. However, conducting a sound Internet measurement study is a difficult undertaking [3][4]. Previous investigations suffer from known limitations, such as limited measurement duration or coverage, loss of information during the measurement process, failure to identify the applications correctly and often the use of a non-recent data traces.

Additionally, the availability of broadband user connections is continually growing; particularly those based on cable TV and ADSL technologies. Such widespread availability opened up new ways of resource usage for both home users and small organizations. With such availability and increased quality of service, users are more inclined to use a wider range of services available in the current Internet, such as Voice over IP (VoIP), e-commerce, Internet banking and P2P systems for resource sharing, particularly of audio and video files. In other words, the increased capacity and availability have led to a complex behavior for a typical user [5], very different from that of a dial-up user. In fact, some recent studies showed that, compared to dial-up users, broadband users get involved with different activities, tend to dedicate more time for creating and managing on-line content and also for searching information [6]. Therefore, Internet Service Providers (ISPs) should pay more attention to such complex behavior, especially in the face of the popularization of recent wireless access technologies such as 4G Long Term Evolution (LTE) cellular systems, fixed and Mobile Wi-Max and Wi-Fi, LANs [7]. Moreover, the current trend of moving phone calls from the circuit switched PSTN to the packet switched Internet using VoIP applications represents a challenge to telephony companies and one that its effects have yet to be completely understood.

To provide greater insights on user network utilization, it is important to investigate and classify the types of network applications that generate user traffic. It is already widely accepted that the use of well-known ports on its own to identify applications is obsolete [8][9][10][11]. This happened for two reasons: many applications do hide themselves as a form of trespassing severe firewall rules by utilizing a commonly open port whereas other applications require the use of dynamic ports as part of their design (e.g., many VoIP protocols). Therefore, alternatives to traffic identification were needed and dozens appeared in recent years. The focus of this survey is to present these alternatives as well as the next steps in the field.

Manuscript received 31 October 2007; revised 7 July 2008. This work was supported in part by Ericsson Brazil under grant UFP.17.

A. Callado, J. Kelner and D. Sadok are with the Federal University of Pernambuco in Recife, PE 50670-901 Brazil (e-mails: {arthur, jk, jamel}@gprt.ufpe.br).

C. A. Kamienski is with the Federal University of ABC in Santo André, SP. 09.210-170 Brazil (e-mail: carlos.kamienski@ufabc.edu.br).

Stênio Fernandes is with the Federal Institute of Education, Science and Technology (IF-Alagoas, Maceió 57.020-510 Brazil (e-mail: stenio.fernandes@ieee.org).

G. Szabó and B. P. Gerő are with Traffic Analysis and Network Performance Laboratory of Ericsson Research Research in Budapest, Hungary (e-mails: {geza.szabo, balazs.peter.gero}@ericsson.com).

Digital Object Identifier 10.1109/SURV.2009.090304.

This paper is organized as follows. For readers new to the field of traffic analysis, chapter 2 presents some fundamental concepts of traffic measurement using traffic mirroring and flow collection and chapter 3 explains some techniques used for traffic analyses. Chapter 4 focuses on some important recent works on traffic analysis for application identification. Chapter 5 compares the results from the most relevant techniques shown in the literature. Finally, chapter 6 makes some final remarks on traffic analysis and lists the open questions and challenges for research in traffic measurement.

II. Introduction

Network traffic measurement has recently gained more interest as an important network-engineering tool for networks of multiple sizes. The traffic mix flowing through most long-haul links and backbones needs to be characterized in order to achieve a thorough understanding of its actual composition. Different applications (traditional ones such as Web, malicious others such as worms and viruses or simply hype such as P2P) affect the underlying network infrastructure. New business and settlement models may be reached between content and transport providers once a clear traffic understanding is achieved. As far as broadband residential users and access providers are concerned, measuring traffic to and from the customer base is essential for understanding user behavior.

In broader terms, measurement strategies can be seen as an essential tool for identifying anomalous behavior (e.g., unexpectedly high traffic volumes, Denial of Service (DoS) attacks, routing problems, unwanted traffic, and so on), for the design and validation of new traffic models, for offering highly demanded services, as well as for helping seasonal activities such as upgrading network capacity or eventually for usage-based pricing. But first, it is very important to differentiate between network measurement and application identification: the former is about data gathering and counting, while the latter is the recognition and classification of some traffic characteristics (which vary according to the technique being used). Traffic identification, however, is inherent to traffic classification, since one may not classify before identification.

According to [3], traffic measurements can be divided in active and passive measurements; and can also be divided in online and offline strategies. In the case of online measurement, the analysis is performed while the data is captured; while in offline measurements, a data trace is stored and analyzed later.

A. Active versus Passive Measurements

Active measurement is defined as measurement obtained through injected traffic. In the case of active monitoring several probe packets are sent continuously across the network to infer its properties. Active measurements are mainly used for fault and vulnerability detection and network or application performance tests. However, it may not be always suitable to reveal network characteristics as influenced by users, due to the fact that active measurement sends packets independently of user behavior and therefore changes the network metrics it is trying to measure in the first place. In addition, network managers may also face scalability issues due to the size of the monitored network. In other words, active monitoring becomes prohibitive due to the large number of prospective end systems that should be tested, as well as the number of experiments that should be conducted in order to gain knowledge about the behavior of a given network. For example, actively measuring available path bandwidth often involves saturating router buffers along it using packet pairs. Most network operators are unwilling to generate extra traffic for active measurement especially when knowing that there is a passive counterpart. Therefore, most commonly used measurement techniques of Internet traffic fall into the area known as passive measurement, i.e., without making use of artificial probing.

Passive measurement is defined as measurement of existing traffic without injecting traffic. Passive techniques are carried out by observing network packets and connections (these are also called flows). A flow is defined as a set of packets that share origin and destination addresses, origin and destination ports (if applicable to the transport protocol utilized), transport protocol and are observed within a time-frame (this is configurable). When using flows as a measurement unit, only traffic summaries of the flows are considered. When observing packets, most techniques work by capturing packet headers and analyzing them. There is an on-going discussion about some legal issues when inspecting packet payload [12][13], e.g. packet payload inspection is forbidden in many countries due to privacy law enforcement. On the other hand, flow-based measurement deals with a summary of unidirectional streams of packets passing through a given router. To achieve an in-depth characterization of network traffic, passive measurements can be undertaken in both levels, namely packet-based and flow-based measurements. Another information source for traffic analysis comes from the use of the Management Information Base (MIB) data, available through the Simple Network Management Protocol (SNMP), currently implemented on nearly any network device that can be managed. It provides coarse-grained, low volume, non-application-specific data. Generally, SNMP is not desirable for collecting meaningful data for traffic analysis, because there is no information on packets (except their total number seen at a given interface, which can be polled for a low-resolution total volume analysis) or flows, and consequently some vital information is lost, such as the endpoint addresses, port numbers, protocol type, etc. It is not possible to try and infer the application based on the data provided by SNMP. This is only appropriate for total volume measurement and per-interface traffic accounting.

Passive measurement techniques are particularly suitable for traffic engineering and capacity planning because they depict traffic dynamics and distribution. Their main limitation however has to do with dealing with massive amounts of data (except for SNMP) known to scale with link capacity and with the size of the user base. Since the main purpose of this survey is to identify traffic but not to manage, monitor, shape or block traffic, only passive measurements are discussed.

The network interface processor has a huge impact on the quality of the measurement and on the possibilities of online classification. For a discussion on the hardware issues of implementing an online classifier, see chapter 12 in [14]. For

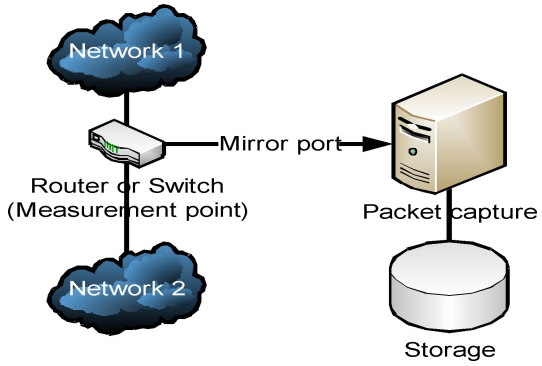

Figure 1. Packet Measurement Topology

a detailed explanation of network processors, please refer to [15].

B. Packet-Level Passive Measurements

At a microscopic level, measurements are performed on each packet traveling across the measurement point. The information collected can be very fine-grained. Examples of relevant collected information are source and destination IP address, source and destination port numbers, packet sizes, protocol numbers and specific application data. There are several packet capture tools (sniffers) freely available, most of which rely on the libpcap library. TCPdump[1] is a commonly used tool that allows one to look closer at network packets and make some statistical analysis out of the trace files. Ethereal[2] [16] or Wireshark, as currently known[3], adds a user-friendly GUI to TCPdump and includes many traffic signatures that can be used for accurate, payload-based application identification. SNORT[4] is a tool for real-time traffic analysis and packet logging, capable of performing content searching/matching and detecting many types of network security attacks. A set of other packet-related tools may be found in the Internet[5].

There are three possible hardware combinations for packet capture. First, one can use a cable splitter (the fiber splitter is very common for that) to capture traffic without affecting the traffic in any way, since no equipment is added to the path of the packets. On a second and similar possibility, a passive equipment is used for port mirroring. In practice, the speed of a mirror port may limit the number of monitored ports, which will not significantly affect the delay of the packets (Fig. 1, with a switch). In the third possibility, an active equipment is put in the path of the packets, and it can also, optionally, act as a firewall or traffic shaper. This equipment will perform traffic capture to a disk or perform traffic mirroring for another machine (Fig. 1, with a router), which may alleviate processing usage. In the example shown in Fig. 1, a machine is directly connected to the measurement router (or switch) using port mirroring. This may increase packet delay, but the equipment will not change the packet's contents. Depending on which part of the captured data will be stored, some processing may be required in the packet capture machine such as converting some fields or hiding certain information for privacy concerns. Next, the data may be stored in a local or remote database for scalability and proper data management and will be available for traffic management analysis requests.

The amount of data needed for storing packet traces is usually huge and often has prohibitive costs. The efficiency in accessing such databases for analyses is also critical. This justifies the use of a Database Management System (DBMS) for data storage. Another common problem of packet capture is that sub-optimal hardware (e.g., the use of a low-end network interface and low CPU power) may affect the packets capture integrity at near full wirespeed, and contribute to packet loss.

C. Flow-Level Passive Measurements

At a macroscopic level, measurements are performed on flow basis. In this case, aggregation rules are necessary to match packets into flows. Collected data include the number of flows per unit of time, flow bitrate, flow size and flow duration. Examples of commonly used tools that deal with flows are Cisco's NetFlow[6] [17] (the de facto standard) and Juniper's JFlow[7].

Cisco was the first to come up with and implement a flow-level capture solution. NetFlow provides a set of services for IP applications, including network traffic accounting, usage-based network billing, network planning, security control, Denial of Service (DoS) monitoring capabilities, and network monitoring. It is currently seen as the most important technology for measuring and exporting traffic flows[8].

Cisco released 6 versions of NetFlow: versions 1, 5, 7, 8, 9 and 10 (also called IPFIX - IP Flow Information Export) [18]. Beginning in version 9, NetFlows' format is configurable and adaptable. Today, however, the most widely used version of NetFlow is version 5[9].

Although NetFlow v5 provides many fields of information, in practice many programs fail to correctly fill all its fields. Consequently, the only systematically utilized (i.e., correctly fulfilled) and therefore dependable fields are: Source IP, Destination IP, Source Port, Destination Port, Layer 4 Protocol, Packet Count, Byte Count, Start Time and End Time. A router configuration[10] must include the timeout for active and inactive flows, which might considerably affect the results, as it breaks a flow into smaller ones.

JFlow also provides a similar set of functionalities and supports NetFlow's export formats. Actually, most software developers of flow collectors along with the leading companies in the router-related industry are working jointly within the IETF to build a standard for flow records representation known as IPFIX [18]. This is largely based on a previous Cisco

[1]http://www.tcpdump.org
[2]http://www.ethereal.com
[3]http://www.wireshark.org
[4]http://www.snort.org
[5]http://www.tcpdump.org/related.html

[6]http://www.cisco.com/warp/public/732/netflow/
[7]http://www.juniper.net/techpubs/software/erx/junose80/swconfig-ip-services/html/ip-jflow-stats-config2.html
[8]http://www.cisco.com/en/US/tech/tk812/tsd_technology_support_protocol_home.html
[9]http://support.packeteer.com/documentation/packetguide/7.2.0/nav/overviews/flow-detail-records-overview.htm
[10]http://manageengine.adventnet.com/products/netflow/help/installation/setup-cisco-netflow.html

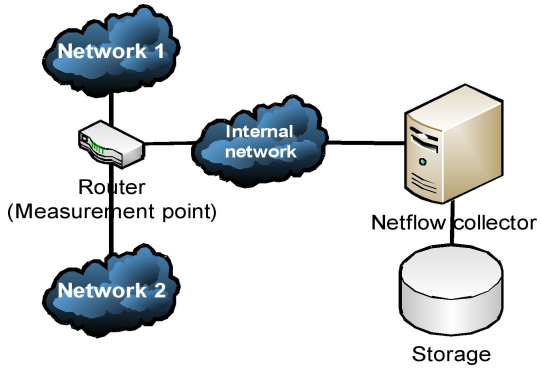

Figure 2. Flow Measurement Topology

proposal, namely, called NetFlow 10. Iannaccone shows [19] that in many cases there is an advantage in generating the NetFlow data using port mirroring. There are good available software tools for working with both NetFlow and JFlow flow traces, such as Flow-Tools[11] and Ntop[12]. Ntop also works as a packet level sniffer, i.e., it has packet capture features.

In Fig. 2, we see the topology of a NetFlow measurement. At this point, the flow collector does not have to be directly connected to the router where the measurement is performed – as this will not affect the quality of the collected data. The collector will feed the database with flow information, and this information will be available for further traffic analysis.

D. Sampling Techniques

Both flow monitoring tools and packet level capture tools suffer from a lack of scalability relative to link capacity, since monitoring high-speed links generates a massive amount of data. As link capacity and the number of flows passing through a router grow, maintaining both individual counters and state information for each flow traversing it becomes computationally intense to keep up with. Therefore, sampling becomes a crucial technique for a scalable Internet monitoring. It is a reasonable statistical strategy for dealing with high volumes of data. It has the advantage of lowering the processing cost, the storage and hardware requirements in a network monitoring infrastructure. Please note that reducing the volume of collected data induces information losses, although such reduction makes the monitoring process highly scalable. In general, sampling network traffic is the process of making partial observations of packets or flow records, and drawing conclusions about the behavior of the system from these sampled observations. In other words, this process aims at minimizing information loss while reducing the volume of collected data. Indeed, such reduction makes the monitoring process scalable for a variety of network sizes. Transforming the obtained partial data into information or knowledge about the overall system behavior is known as the inversion problem.

A variety of sampling strategies have been proposed recently for optimizing the packet selection process (for flow accounting) [20][21] and flow selection (for statistical analysis of the original traffic) [22]. Sampling may be applied during the capture (packet level, as is done on CISCO routers[13]) or after data classification and aggregation (flow level). Note that sampling flow measurement has become increasingly employed at packet level in order to reduce the computational overhead found in high-speed backbone routers. Sampling techniques may be divided into systematic, random and stratified sampling. The simplest sampling process is uniform 1/N sampling (systematic sampling), where the first one of every N packets is retained. Although this technique has been largely used, e.g. by NetFlow in high-speed links, it does not always present adequate results since it has been shown that IP flows usually have heavy tailed distributions for packets and bytes [22]. This results in a bias towards smaller packets and flows. Another very simple form is random 1/N sampling (random sampling), where a random packet of every N packets is retained. It has nonetheless the same problem; the average traffic volume estimated with random sampling is biased towards a smaller average value. Henceforth any traffic monitoring technique based on uniform or random sampling will provide an underestimation of traffic volume.

It is advocated in [23] that reducing the number of collected traffic flows (regardless of the previous use of packet sampling during the actual packet capture), may be undertaken by applying stratified sampling to them. This technique is based on the idea of increasing the estimation accuracy by using some a-priori information. First, it performs an intelligent grouping of the elements of the parent population. Many levels of grouping may be considered. The samples are then taken from each subset or stratum. When subpopulations vary considerably, e.g. in traffic flow size and duration, sampling each subpopulation (stratum) independently may be advantageous for reducing the sampling error. Also in [23], the authors use flow sizes to divide the parent population and estimate both volume and duration of flows. They show the effectiveness of using stratified sampling as a tool for describing traffic behavior at a flow level. A decisive step is achieved by conducting an accurate statistical population study, which affects heavily the way the strata are created. A straight consequence of this approach is a considerable reduction in the number of flows needed for computing descriptive measures that are highly representative of the original non-sampled flows records. The measurement process is reduced to the capture and storage of only a small fraction of the original number of flows. The results of the conducted evaluation show that the use of stratified sampling provides significant and promising advantages when applied to network traffic flow monitoring.

It is worth stressing that sampling is a lossy process but may have significant impact on the processing times at both routers and collectors. It may also have an unexpected impact on the traffic classification techniques that use volume or behavior information. Therefore, it should be applied carefully in order to avoid unacceptable inaccuracy or unnecessary processing overhead. For example, in the research work presented in [24], the authors perform a study on the accuracy and overhead of the NetFlow's sampling feature while running over a Cisco GSR 12000 series router. In their experiments, they turned

[11]http://www.splintered.net/sw/flow-tools/
[12]http://www.ntop.org
[13]http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_ feature_guide09186a00801a7618.html

on the Sampled NetFlow with a systematic sampling of 1-in-250. They also passively captured packets from the same flows in order to evaluate the accuracy of NetFlow. During the experiments, the CPU and memory utilization stayed low and were mainly requested by routing related processes. They found that bandwidth usage due to the export of packets grows linearly as the number of flows increases. Another result was that NetFlow accurately sampled packets at the previously configured rate. Finally, they showed that the performance of both systematic and random sampling is similar.

III. State-of-the-Art in Flow analysis

As we pointed out earlier in this article, one of the main problems with passive measurement is dealing with a massive amount of data, since the volume of captured data can become very large on high-capacity links. Additionally, the network manager should make important design decisions on how to cope with different granularity levels in order to gather useful information for network traffic engineering. Essentially, there is a broad avenue for future research in this field, spanning all the way from defining strategies – to sampling - for dealing with the huge amount of network traffic data. In addition, obtaining an in-depth knowledge of the trade-off related to the granularity of traffic measurement in a major ISP remains an open topic.

In [25], Liu et al. argue that there are many challenges in the context of network monitoring and measurement. For example, a management approach may address how much information is included in traces and the maximum allowable compression level for such traces. The authors try to find how much information is captured by different monitoring paradigms and tools including full packet header measurement, to flow-level captures and to packet and byte counts (e.g., as with MIB/SNMP). Additionally, they evaluate how much joint information is included in traces collected at different points. Therefore, in order to address these issues properly, the authors developed a network model and an information theoretic framework. In [26] Liu and Boutaba describe a P2P system, called pMeasure, which is capable of creating a large number of measurement nodes on the Internet and providing a synchronized and cooperative measurement system. The authors argue that the system can accept measurement tasks, locate required measurement facilities and fulfill these tasks. All experiments and simulation results point out that p2p-based measurement system are very promising, but no results of a practical measurement were shown.

As an important step for a cautious deployment of any new technique for network management, the research work in [24] analyses the loss of information caused by TCP (Transmission Control Protocol) traffic aggregation in flows. A tool called FLOW-REDUCE was implemented to reconstruct TCP connection summaries from NetFlow export data (named flow connections). The problem is that NetFlow breaks flow information in 5-minute flows, thus possibly breaking a given captured flow into many flows. To study how accurately information is produced by FLOW-REDUCE, TCP connection summaries are reconstructed from packet traces using the BRO tool [27]. When analyzing the differences between flow connections and packet connections, the FLOW-REDUCE heuristic makes a good match for long-lived connections, more specifically, those longer than a router's inactive timeout parameter. However, for connections that have smaller durations, flows and packets connections differ considerably. The authors state that this happens because NetFlow aggregates short-lived TCP connections. The next section details studies in flow duration and other flow characteristics.

A. Classification: Volume, Duration, Rate and Burstiness

In a similar approach to [24], the research work in [28] performs a multi-scale and multi-protocol analysis to explore the persistency (volume, duration, rate and burstiness) properties of those flows that contribute the most to bandwidth utilization (the flows are called elephants or heavy hitters). The authors argue that knowing the persistency features of heavy hitters and understanding their underlying causes is crucial when developing traffic-engineering tools that focus primarily on optimizing system performance for elephant flows. The main difficulty that arises when studying the persistency properties of flows is that the available measurements are either too fine-grained to perform large-scale studies (i.e., packet-level traces) or too coarse-grained to extract the detailed information necessary for the particular purpose (i.e., NetFlow traces or MIB/SNMP data). They also checked the validity of the assumption that flows have constant throughput through their lifetime, by comparing NetFlow-based findings against those obtained from the corresponding packet-level traces. By considering different time aggregations and flow abstractions (e.g., raw IP flows, prefix flows), varying the definition of what constitutes an elephant, and slicing the traces according to different protocols and applications, the authors present a methodology for studying persistency aspects exhibited by Internet flows. The authors conclude that in general using aggregation (NetFlow) for observing tendencies is fine, but heavy hitters should be observed closely (using a packet trace). They also conclude that heavy hitters stay as heavy-hitters for most of their lifetime and mice rarely become heavy-hitters.

Using similar arguments to [29], Lan and Heidemann [30] argue that understanding the characteristics of Internet flows is crucial for traffic monitoring purposes. They first show that several prior research studies on Internet flows have characterized them using different classification schemes. For instance, one can focus on the study of the "elephants and mice phenomenon". It is well known by the Internet community that a small percentage of flows are responsible for a high percentage of the traffic volume. Therefore, identifying elephant flows plays an important role for traffic engineering. In broader terms, one can find flow classification according to size (small sized flows are called mice [31]), by duration (where longer flows are called tortoise and shorter ones are called dragonfly [31]), by rate (heavy flows are called cheetah and light ones snails [30]) and by burstiness (bursty flows are called porcupine and non-bursty ones stingray [30], but these are also called alpha and beta traffic). As a rule of thumb, they adopted a classification whereby a flow in a trace is considered an Elephant if its size is bigger than the average size (for the trace) plus 3 times the standard deviation of the size (for the trace). A flow in a trace is seen as a tortoise if its duration

is longer than the average duration plus 3 times the standard deviation of the duration. Similarly, a flow in a trace is a cheetah if its rate is higher than the average rate plus 3 times the standard deviation of the rate. Finally, a flow is labeled as a porcupine if its burstiness is higher than the average burstiness plus 3 times the standard deviation of the burstiness. However, at first it is not clear how these different definitions of flows are related to each other (correlation). The authors consider it important for network administrators to be able to identify the flows and the rates of flows that are more troublesome to routers (big, long, heavy, bursty or combinations).

By relying on data recorded from two different operational networks (Los Nettos and NLANR) Brownlee et al [32] studied flows in four different dimensions, namely size, duration, rate and burstiness, and examined how they could be correlated. This research analyzes the relationships between the different heavy-hitters and concludes that there is a strong (over 80%) correlation between flow rate and burstiness, between flow size and rate and between flow size and burstiness, while there is a small correlation between flow duration and all the other metrics. A few of the exceptions seem to be related to network attacks (multiple TCP SYN retransmissions), which seem to be a good proposal approach for a future work to identify network attacks based on the detection of these types of flows. In summary, they made three main contributions: First, they characterized prior definitions for the properties of such heavy-hitter traffic. Second, based on the available datasets, they observed strong correlations between some combinations of size, rate and burstiness. Finally, they provided an explanation for such correlations, by relying on transport and application-level protocol mechanisms.

With data in hand, it is crucial a network operator to gather information from the available network traffic records. In [31] the authors present three techniques for the identification of elephant flows. The "aest" technique considers the flow-bandwidth distribution as a heavy-tail and sets the threshold in such a way to isolate the tail of the distribution, labeling the flows in the tail as elephants. The "α-constant load" technique needs an input parameter α, corresponding to the percentage of traffic that will be labeled as elephant. These first two single-feature techniques (aest and α-constant load) result in highly volatile elephant classification over time. A third technique periodically calculates the distance between a flow rate and the threshold value calculated with the aest and α-constant load techniques and sums the last 12 calculations to obtain the "latent heat" metric for stability. Its authors argue that such approach is more successful in isolating elephants that exhibit consistency – high volume over long time.

B. Traffic Characterization

Many network traffic modeling research papers initiate with a traffic analysis approach before proposing any analytical one. Therefore, one can take advantage of this procedure to gain some important knowledge on the most common types of analysis for network traffic. For example, many network traffic modeling studies aggregate all connections together into a single flow. It is well known that such aggregate traffic exhibits long-range dependence (LRD) [33][34] correlations and non-Gaussian marginal distributions. LRD traffic signals that the traffic exhibits only small variations on the intensity of self-similarity over different timescales. In the context of internet measurement, it is important to know that in a typical aggregate traffic model, traffic bursts arise from several simultaneously active connections.

In [35], the authors developed a new framework for analyzing and modeling network traffic that reaches beyond aggregation by incorporating connection-level information. A careful study of many traffic traces acquired in different networking situations reveals that traffic bursts typically arise from just a few high-volume connections that dominate all others. In that paper, such dominating connections are called alpha[14] traffic, which is caused by long transmissions over high bandwidth links and is sometimes extremely bursty (non-Gaussian). Stripping the alpha traffic from an aggregate trace leaves a beta traffic residual that is Gaussian, LRD, and shares the same fractal scaling exponent as the aggregate traffic. Beta traffic is caused by both short and long transmissions over low bandwidth links. In their alpha/beta traffic model, the heterogeneity of the network resources gives rise to burstiness and heavy-tailed connection durations leading to LRD. Queuing experiments suggest that the alpha component dictates the tail queue behavior for large queue sizes (i.e., bursty traffic makes big queues fill up), whereas the beta component controls the tail queue behavior for small queue sizes (i.e., constant-rate traffic will not affect queue size, except when queues are very small). The potential causes of burstiness might range from the transient response to re-routing, the transient response to start/stop of connections, the TCP slow-start peculiarities to the heterogeneity in bottleneck links for passing flows.

Traffic modeling in terms of packet distribution and packet burst distribution according to application type is useful for traffic load studies but not relevant for application identification as opposed to network load/performance analysis and traffic engineering [36]. This is due to many applications sharing the same timing characteristics, which results not only from application behavior, but also from commonly shared TCP's congestion avoidance mechanisms and router queuing policies. For more information on traffic modeling and its issues, see [34][37][36].

In a general analysis of Internet traffic, Kim et al [38] performed a flow-based investigation on four 1Gbps network links from an academic network. They discovered a frequent occurrence of flash flows, which may affect the performance of the existing flow-based traffic monitoring systems. There are some interesting results from this paper. First, they confirm that the relation between the number of bytes and the duration of a flow is very weak, i.e., non-correlated. Moreover, by analyzing flow count over time, they conclude that flash flows are mostly responsible for flow count fluctuation over time. Therefore, the paper concludes that ignoring flash flows will eventually improve the performance and accuracy of flow/volume based prediction systems.

The concept of flow aggregation through Internet links is presented in [32]. This interesting work describes a method

[14]Although the name is equal to the alpha/beta flows previously cited, the definition is different, so it is important not to confuse them

of measuring the size and lifetime of Internet flows based on the use of the NeTraMet[15] tool (Network Traffic Flow Measurement Tool). From the analysis of the available datasets, the authors find that although most flows tend to have a short lifetime (therefore, are called dragonflies), a significant number of flows have lifetimes lasting as long as several hours and even days, and can carry a high share (10-25%) of the total bytes on a given link. They also define tortoises [32] as flows that last longer than 15 minutes (representing 1-2% of the traffic they observed). They point out that flows can be classified not only by lifetime (dragonflies and tortoises) but also according to size (mice and elephants), and observe that flow size and lifetime are independent dimensions. Following previous studies, the authors argue that ISPs need to be aware of the distribution of Internet flow sizes, and the impact of the difference in behavior between short and long flows. In particular, they advocate that any forwarding cache mechanisms in Internet routers must be able to cope with a high volume of short flows. Moreover, ISPs should realize that Long-Running flows can contribute a significant fraction of their packet and byte volumes – something they may not have allowed for or predicted when using traditional 'flat rate user bandwidth consumption' approaches to provisioning and engineering.

Following the same approach of his previous work in [32], Brownlee [39] analyzes network flow lifetimes for a backbone link at two different sites. He studies the effect of the flow capture strategy that involves discarding the short-lived flows, referred to as dragonflies. Long-lived ones are called tortoises. The author observed that a high proportion of traffic bytes are carried by tortoise flows. Brownlee suggests that ignoring flows with six or fewer packets results in the long term of about only 2% of "user" traffic being ignored, but greatly reduces the number of flows and the flow processing. Therefore, this technique may permit the use of flow monitoring on faster links, up to 1 Gbps, without the need for dedicated hardware.

C. User Behavior

Due to the increased complexity and processing power required for performing user behavior analysis from packet traces and the restrictions on payload data usage in force in many countries, recent studies focus on the analysis of flow traces or connection-level behavior. Effective traffic analysis will provide statistically sound general network profiles and application-specific behavior.

Volume analysis has already been considered a very insensitive method for anomaly detection, although it may reveal few anomalies faster and easier. This is due to the aggregative and consequently information destructive characteristic of volume statistics. Packet-level analyses are even forbidden in some countries, and therefore can be used only when necessary and permitted. Flow-level analysis saves on processing resources and has also shown to be useful for anomaly detection. Flow and volume analysis together should form a good methodology for anomaly detection, considering that both detect rather disjoint sets of anomalies. Furthermore, some previous work has been done on IP traffic characterization and focused on understanding statistical properties of packet and flows at the network and transport layers. In this area some works are considered understanding seasonal traffic volumes, user connection durations, traffic growing trends, packet arrival processes, self-similar [40] (fractal) behavior and traffic matrix estimation. This crucial information has been used both by ISPs for network dimensioning and resource provisioning and by the Internet research community for an in-depth understanding of the current Internet traffic state and protocol design. At the time being, a variety of network traffic information is provided by tools such as NetFlow and related flow processing software. For residential and SOHO (Small Office, Home Office) customers, a preliminary characterization of user behavior can be found in [6][5].

There are some relevant metrics to grab user behavior which have been studied in recent years. User data volume, session arrival and session duration are the most utilized in [5][6][7], but application-specific quality metrics are also relevant, though harder to measure. The measurement of application-specific metric requires access to the user machine, which explains its seldom use. As an example, an analysis of VoIP application quality metrics can be seen on [41].

To conclude this section we describe two recent research studies that evaluate broadband user behavior. It is worth stressing that while there are many papers that address Internet measurements, there are few research papers which provide an actual in-depth traffic analysis of broadband users.

In [42], Sinha et al studied Internet customer traffic behavior by comparing the upstream links of two different "last-mile" technologies, namely Broadband Fixed Wireless (BFW) and Digital Subscriber Lines (DSL). Unsurprisingly, given the preponderance of downloads over uploads, their analysis (using packet traces and NetFlow data) showed that most flows (created by NetFlow) in the uplink are short-lived for both access networks and indicate that this is mostly due to download TCP ACKs (mainly HTTP and P2P traffic), other TCP control packets (SYN, RST, FIN) and to DNS requests. They found out that 80% of the flows are equal to or less than 10 seconds in the BFW and 3 seconds in the DSL. Almost 30% of the upstream flows in BFW and nearly 38% of the upstream flows in DSL consisted of single packets smaller than 100 bytes. However, the increasing use of P2P applications for file uploads increases the average flow duration significantly. The analysis of flows inter-arrival times shows a near-range correlation for DSL and a significant burstiness for BFW, which is believed to occur due to the fact that in BFW bandwidth is shared among users, while in DSL all links are dedicated to their users. This work can be seen as a first step to construct a generalized parametric model for broadband access networks.

In [6], Marques et al first separate users' traffic in two main categories, namely residential and business. They analyzed user session arrivals and durations and concluded that these follow exponential and lognormal processes, respectively, for both residential and business users. They also analyzed the profile of different types of applications, such as Web, Peer-to-Peer, Instant Messaging and POP3. In addition, they pointed out that business users tend to stay connected longer than residential ones, while residential users tend to connect

[15]http://www.caida.org/tools/measurement/netramet/

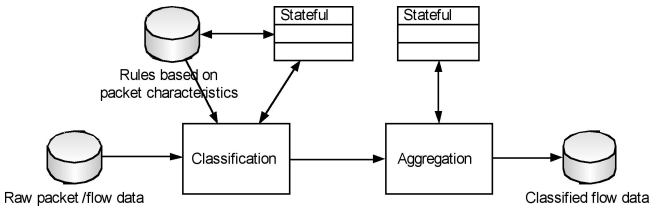

Figure 3. Packet Classification Methodology

more often. As expected, residential user session arrival was distributed over the day, while business user session arrival was biased towards office hours. Another contribution is the proposal of a Customer Behavior Model Graph (CBMG), which consists of a state transition graph, divided in classes, that tries to model the user behavior, specifically user request patterns.

IV. STATE-OF-THE-ART IN INTERNET TRAFFIC CLASSIFICATION

Traffic classification can be performed using either packet or flow data. Fig. 3 shows the packet classification methodology.

Here, the captured packets are classified based on packet level characteristics or application signatures. Note that packet level application classification may be a stateful process. After that packets are aggregated in flows (for reliability and scalability) before storage. The traffic signatures must be created before-hand and require the work of a specialist. They must also be updated frequently to readapt to emerging applications.

Flow classification can be of many types. Some classifiers infer the application simply by the ports used. Others match flow level characteristics to predefined flow characteristics using e.g. connection patterns. These use data structures that describe an application behavior (e.g., graphlets or attribute entropy) to compare with observed attributes. A third type uses machine learning techniques (e.g., bayesian networks, statistical analysis) to infer the application based on previously classified applications. After classification, the flow information is stored in a database along with the classification result. In the next section, the main metrics for traffic identification are presented.

A. Inference Quality Metrics

To evaluate the quality of any measurement based on inference, it is very important to have a trustable classification reference or baseline. This reference may be accomplished by the injection of synthetic traffic (where all flows are previously identified since the applications that generated them are known), by hand-made classification (an analyst looking at the whole traffic trace) or by a trustable packet analyzer (previously validated with one of the other two methods). The two most used inference quality metrics are completeness and accuracy. Fig. 4 illustrates the idea behind these metrics.

Fig. 4 (a) shows the actual separation between a desired application (A) and all other applications (O). Fig. 4 (b) shows the same traffic, but with a perceptive separation between what was detected as the desired application (D) and what was not detected as the desired application (U). Fig. 4 (c) shows the intersection of both real and perceptive views, making clear what was correctly detected (AD), what was incorrectly detected (false positives - FP), what was incorrectly undetected (false negatives - FN) and what was correctly undetected (OU).

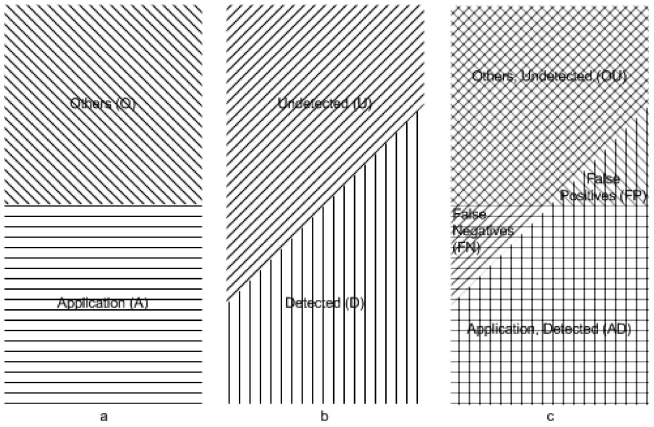

Figure 4. Application Detection Metrics

Accuracy signals how correct a detection technique is. Detection accuracy is measured in percentage (ratio between correct detections and total detection count) and may not be more than 100%. The lack of accuracy leads to false positives, e.g.: when some inference technique tries to detect all flows that correspond to a certain application and it detects flows of other applications as belonging to the desired one, these extra detected flows are false positives and their presence diminishes the value of accuracy. Given the units described above, the accuracy is calculated as:

$Accuracy = \frac{AD}{D}$

Completeness is a metric meaningful over the coverage of a detection technique and may be computed in relation to flows, bytes or packets. Detection completeness is measured in percentage (a ratio between the detected count and the real application count), and may be over 100% (when more flows, bytes or packets are detected than what should be). Low detection completeness indicates the presence of many false negatives, e.g.: when some inference technique tries to detect all flows that correspond to a certain application and it does not detect all flows belonging to that application, these undetected flows are false negatives. Given the units described above, the completeness is calculated as:

$Completeness = \frac{D}{A}$

It is important to notice that both completeness and accuracy metrics are independently calculated and mutually complementary. An inference technique may have 100% completeness and have a very low precision, and vice-versa. To be considered useful, a technique must have a high completeness and a high accuracy (authors believe both being over 90% gives reasonable results).

B. Goals on Traffic Classification

Accurate traffic classification is fundamental to numerous network activities [43][44], including:

• Identification of application usage and trends: the correct identification of user applications and application popularity trends may give valuable information for network operators to help traffic engineering and for providers to offer services based on user demand.

• Identification of emerging applications: accurate identification of new network applications can shed some light on frequent emergence of disruptive applications that often rapidly alter the dynamics of network traffic, and sometimes bring down valuable Internet services.

• Anomaly detection: diagnosing anomalies is critical for both network operators and end users in terms of data security and service availability. Anomalies are unusual and may cause significant changes in a network's traffic levels, such as those produced by worms or Denial of Service (DoS) attacks.

• Accounting: from an ISP perspective, knowing the applications its subscribers are using may be of vital interest for application-based accounting and charging or even for offering new products. For example, an operator may be interested in finding out users who are using voice (VoIP) or video applications.

C. Port-Based Application Deduction

The most common traffic classification method maps port numbers to applications, i.e. an application is associated with a known port number [45]. This method uses only packet headers (or flow identifiers).

Most often, the classification is based on associating a well-known port number to a given traffic type, e.g., web traffic is associated with TCP port 80 [45]. This method needs access only to the header of the packets.

The main advantage of port based method is being fast as it does not apply complex calculations. The implementation of a port based classification system is quite simple and the knowledge of the classification system can be easily extended by adding new application-port pairs to its database. For common services e.g., DNS (53), FTP (20, 21), e-mail (25, 110, etc.) it works well.

Nowadays, networks exceedingly carry more and more traffic that uses dynamically allocated port numbers. Further, many applications choose to hide their traffic behind for example HTTP traffic carried over TCP port 80, in order to get through firewalls. As a consequence, the port based method becomes insufficient in many cases, since no specific application can be associated to a dynamically allocated port number, or the traffic classified as Web may easily be something else carried over TCP port 80. It is also quite common that a specific application uses a non-default port number, which results in the failure of the port based method.

D. Packet-Based Analysis

The most accurate solution is obviously complete protocol parsing. However, there are many difficulties tied to this method. Some protocols are designed to be secure, thus their data stream is ciphered, i.e. intentionally hidden from sniffer programs. Another problem is that for proprietary protocols, there is no publicly available protocol description. Furthermore, it is a great deal of work to implement every protocol parser that might occur in a network. In addition, to parse all protocols for all users separately is a computationally complex and unscalable task. Finally, some countries also have laws prohibiting network operators from parsing the contents (payload) of network packets, which would be a privacy violation in their view. Due to these difficulties, complete protocol parsing is not a valid solution in itself, but it may be used in combination with less computationally complex methods.

One way to make traffic classification less resource consuming is to search for specific byte patterns – called signatures - in all or part of the packets in a stateless manner. This heuristics based approach uses predefined byte sequences or signatures to identify particular traffic types, e.g., Web traffic contains the string '\GET', eDonkey P2P traffic contains '\xe3\x38'. The common feature of the signature-based methods (also known as payload based methods) is that they look into packet payloads in addition to packet headers. One problem with signature based methods is that it is difficult to maintain signatures with high hit ratio and low false positive ratio and many times only real experiences with traffic traces provide enough feedback to select the best performing byte signatures. In order to illustrate, the above example with the '\GET' signature finds both HTTP and Gnutella, thus it is not adequate for accurate traffic classification. Another disadvantage is that this method cannot deal with encrypted content. A further issue arises when signature based analysis is run off-line: packets are often recorded with a limited length (e.g., 200 bytes), in order to reduce the size of trace files. Thus, it might happen that the recorded payload part does not contain the signature, despite that the original payload contained it. Packet fragmentation may also add to the computational complexity of such methods. It is generally understood that signature based methods are able to perform fairly accurately, but due to the fact that they are based on heuristics, their results cannot be taken for granted.

In [46], Sen et al provide an interesting investigation on payload-based P2P traffic classification. They analyzed the most common P2P applications using application-level signatures. Their technique achieves less than 5% false positive and false negative ratios in most cases. Their application, however, requires previous knowledge of each application for the development of the signature and, therefore, will not automatically adapt to new/emergent applications. Furthermore, many countries forbid the inspection of any packet payload in transport/backbone networks, making this technique undesirable for any international backbone as well as any backbone belonging to such a country.

To ease the manual signature construction process, some approaches focus on extracting application signatures from IP traffic payload content automatically [47][48][49]. All of these methods need a reference packet level trace and reference traffic classification to startup with in the first place.

Another question of interest is the efficiency of signature matching. Basic regular expression checks are very processor intensive and hence slow. There are papers that focus on this problem [50] and use promising techniques for pattern matching such as the application of bloom filters [51].

In statistical based classification some statistical feature of the packet-level-trace is grabbed and used to classify the network traffic. For example, a sudden jump in the rate of packets generated by a host might be the sign of worm propagation. However, a jump in the rate of packets might also be an indication of a P2P application or BGP updates. Some P2P applications are known to generate plenty of zero payload flows while peers try to connect to each other. In case of statistical approaches it is feasible to determine the application type, but specific application/client type cannot be determined in general: e.g., it cannot be stated that a flow belongs to Skype or MSN Messenger voice traffic specifically but it can be assumed that it is the traffic of some kind of VoIP application, which generates packets with constant bitrate in both directions. These flow characteristics can be hardcoded manually or in another way to automatically discover the features of a specific kind of traffic. One research study performs a statistical identification of application type based on traffic attributes and labels it to a specific class of service in the network [52]. In order to achieve better results, statistical methods are combined with methods coming from the field of artificial intelligence. The most frequently discussed method is the Bayesian analysis technique as in [44][53], but some papers compare different methods [54][55][56]. For some papers regarding Skype detection ongoing attempts, see [57][58]. For the automatic extraction of traffic signatures based on connection statistics, see [59].

The work by Moore and Zuev [44] uses Bayesian Analysis to classify flows. In order to perform a statistical analysis of the traffic data using the Bayes technique, appropriate input is needed. To do this, network traffic that had been previously hand-classified provides the method with training and testing data-sets. To evaluate the performance of the Bayes technique, a dataset part is used as a training set in one turn and evaluated against the remaining dataset, allowing computation of the accuracy of classification. The training and testing volume ratios are typically at least 1:1. This is the most problematic in these methods: a lot of previously hand-classified data is needed to classify the rest of the trace. On every different type of traces where data volumes, transport speeds, traffic mix varies the methods may have to be retrained.

In this important work, Moore and Zuev utilized only semantically complete TCP connections (flows that show connection set-up and tear-down: SYN/ACK and FIN/ACK packets). Performing such a pre-processing of a trace is sometimes called trace normalization. The discriminator utilized for the Bayesian analysis included the TCP ports, flow duration, packet inter-arrival time statistics, payload size statistics, effective bandwidth calculation based on entropy and the Fourier transform of the packet inter-arrival times.

Their results (presented in section V) show that the technique is adequate for the classification of most of the applications studied. However, the need for flow lifetime parameters makes this technique unreliable for online classification, since one may not block or apply Quality of Service (QoS) treatment on a flow after it is finished.

Bernaille et al [60][61][62] perform traffic classification based on packet headers. They only use the first few packets of a connection to classify, based on clustering. The packet header trace technique gives more information than the flow trace without inspecting payload.

In their interesting and relevant analysis, known as "On the Fly" classification, Bernaille et al [61] utilized only TCP connections and had access to the first 5 packets (in both directions) of a connection. Although a full trace from the same network is required for training, this approach allows the unsupervised online classification of traffic, and therefore allows actions to be taken during the connections.

This methodology employs data clustering using only the sizes and directions of the first 5 significant packets of a connection. The direction is used as the signal (first and same-direction packets: positive; opposite direction packets: negative). By ignoring TCP handshake packets and ACKs with no payload, the clustering of only 5 packets allows the online classification of flows with a considerable hit ratio for the analyzed applications (see section V). Considering that this approach does not rely on flow volume or duration and that no summarization is required for the analysis, this technique is appropriate for online classification.

A couple of different approaches have been proposed in the literature, but none of them performs well for all different application traffic types present on the Internet. Thus, a combined method that includes the advantages of different approaches is proposed in [63], in order to provide a high level of classification completeness and accuracy. The pros and cons of the classification methods are discussed based on the experienced accuracy for different types of applications. The classification method in [63] is based on the classification results of the individual methods and uses a complex decision mechanism to conclude the end result of the classification. As a consequence, the ratio of the unclassified traffic becomes significantly lower. Further, the reliability of the classification improves, as the various methods validate the results of each other.

Accuracy can be increased but as long as any heuristic is used for traffic classification, the result cannot be exact. Thus, the users of the traffic classification method have to accept to live with minor classification errors.

E. Flow-based Classification

Classifying applications based on flow-level data with the same level of accuracy is challenging, because flow-based application classification are based on less detailed input in comparison their packet based counterparts. Concerning application behavior, one should analyze which constraints apply for application classification, thus making the classification feasible. After analysis of a number of research papers, the most relevant methodologies rely on the use of special knowledge or statistical analysis.

In [43], Karagiannis et al present an important seminal and novel approach to classify traffic flows according to application groups, e.g. P2P, based on connection patterns. Connection patterns are described by special graphs (called graphlets), where nodes represent IP address and port pairs and edges represent flows between source and destination nodes. An example graphlet for the recognition of Web and Game applications is shown in Fig. 5.

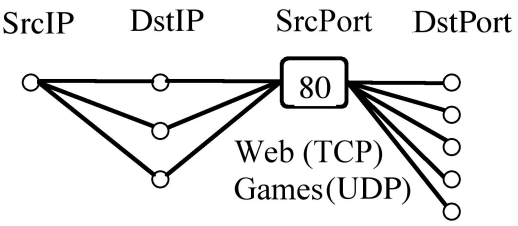

Figure 5. A typical Blinc graphlet describing Web and Game traffic

Karagiannis et al analyzes connection patterns in three levels of detail: the social, the functional and the application level. The analysis performed by proposed the BLINC approach makes use of only the fields belonging to the 5-tuple (source and destination addresses, source and destination ports and protocol) and two other fields: flow duration and size. In the social level, only the addresses are used: it shows how many hosts does a host connect to. In the graphlet shown in Fig. 5, one host connects to many hosts, which means this host is very popular.

In the functional level, the source and destination ports are used. However, it is done with no knowledge of standard port numbers. In the example, the left node is identified as a server, since many hosts connect to it in the same ports but each of the other hosts use a different port.

Finally, in the application level, the type of application is perceived through the relationship among these four fields and, if necessary, the protocol, the flow duration and the flow size (bytes or packets). The calculation of average packet size may be also used. It is relevant to notice that BLINC uses no additional information other than what current flow collectors provide. On the other hand, connection patterns require a large amount of flow data and finished flow lifetime to perform the analyses. Therefore, the BLINC technique is suited to the offline identification of traces of multiple flows (aggregation).

In [29], Xu et al. present an important work that provides a general methodology for building comprehensive behavior profiles of Internet backbone traffic in terms of communication patterns of end-hosts and services. This methodology employs four steps: data preprocessing, significant cluster extraction, cluster behavior classification and behavior classes' interpretation.

In the first step, the data is aggregated into 5-tuple flows and these flows are grouped into clusters. In the next step, the significant clusters are extracted. This is done by checking the size of the clusters and checking if the remaining (non-significant) clusters appear to be random.

In the third step, using entropy and relative uncertainty calculations, the cluster behavior is classified into behavior classes. In this phase, the similarities of clusters are checked and the distributions of addresses and ports are analyzed using entropy. Theses distributions are summarized by relative uncertainty (how much each variable changes in a cluster) into behavior classes.

As a last step, Xu et al use a structural modeling of dominant activities to attribute the resulting behavior classes to known applications. This is accomplished by performing a dominant state analysis, where the dominant combinations of values of the 5-tuple are measured and stored for each behavior class. Using data sets from the core of the Internet, they demonstrate that the methodology can identify common traffic profiles as well as anomalous behavior patterns. An improvement of this technique is shown in [64]. This technique does not rely on volume information and therefore does not need finished flow lifetime. However, due to clustering, there is a need for a multiplicity of captured flows for the technique to be effective, which limits the online usability of the technique on many network links. But this limitation is surpassed by the use of training. In this case, the algorithm trains with different sets of applications and then, in the classification phase, does not need the multiplicity of flows to form clusters to be able to classify applications, since these were defined during training.

The definition of what is unwanted traffic is still very unclear and greatly varies among networks, especially from networks in different countries with different legislations [65]. It is however generally accepted that unwanted traffic is not only about illegal activities, but also about legal activities that affect other users and services. The activities generally cited as unwanted traffic include [65]: Spam, Denial-of-Service (DoS) attacks, Distributed DoS attacks (DDoS), host misconfiguration, host scanning and host exploiting.

While focusing on anomaly detection (not application identification), Soule et al [66] use Origin-Destination flows to generate a traffic matrix and compare four tests (the last two new proposals) to detect anomalies, namely, threshold, deviation score, wavelet transform and generalized likelihood ratio. They use flows collected from Abilene and also synthetically generated traces to validate the proposal, choosing the first as the best one. There are other proposals [67][68] that also try to identify anomalies in traffic using wavelets.

By arguing that the challenge of effectively analyzing the massive data source for anomaly diagnosis is yet unmet, Lakhina et al [69] advocate that the distributions of packet features (IP addresses and ports) observed in flow traces reveals a wide range of anomalies. Using entropy as a summarization tool, they show that the analysis of feature distributions enables highly sensitive detection of a wide range of anomalies and enables automatic classification of anomalies (but not application) via unsupervised learning. They also show that anomalies naturally fall into distinct and meaningful clusters, which can be used to uncover new anomaly types.

V. Traffic Classification Results

When studying a traffic classification technique with real traces, it is important to have a baseline for traffic classification that will be used as a trustable reference. This can be achieved by manual classification of traffic traces, by the use of active measurement [70] or by another method (e.g., a payload classification tool) that was proved to have a high accuracy. It is not yet clear how recent this proof should be, since new applications keep emerging daily. Due to similar reasons, it is essential to validate the classification approaches with measurements from more than one network.

Another option would be to generate the traffic that will be analyzed, therefore knowing the exact applications, but this

also requires a longer time and effort and does not allow a comparison with real traffic as there is no guarantee that results will correspond to those from a real network.

Due to the problems discussed above, validation remains a difficult task. Comparing the accuracy and completeness of the algorithms based on different measurements with potentially different reference classifications, over different networks is not straightforward. In addition, some papers only evaluated the accuracy and not the completeness of their methods. In addition, each paper uses a different traffic source for evaluation, which makes a trustable comparison unreliable. The next section presents a comparison of different traffic identification methods based on the results of their authors.

A. Comparison of traffic identification methods

The results of the BLINC [43], the Bayesian [44], the "On The Fly" [61] and the Payload Analysis [63] methods are summarized in Table I.

With BLINC [43], the authors develop their own byte signatures to establish a baseline and use it for validating the proposed connection pattern based classification method. As shown in Table V 1, the Blinc algorithm is able to recognize the main application types. For their trace, the algorithm performs better in terms of accuracy than completeness (conservative detection).

For comparison, the next method we study is the Bayesian Analysis [44]. First, this method needs to be trained with a data set that was previously classified e.g. manually. Then, the method is tested on a different data set. The authors investigate the accuracy of the approach but do not address completeness, which is very important to validate the relevance of the accuracy metric, as explained before. In this method, also shown in Table V 1, the accuracy of the P2P file sharing traffic is much lower than of other applications. This is in line with the fact that P2P applications are diverse and their main characteristics are difficult to grab, especially with the packet metrics utilized by the Bayesian method.

The method proposed by [61], the "On the Fly" algorithm, uses flow clustering and also uses learning for cluster labeling, but it reads only the first few packet headers in each connection. The accuracy of the classification methods is also summed up in Table V 1. Here, the authors use payload analysis as a reference or baseline. According to the results, the On the Fly method works roughly as accurately as the Bayesian method even though it relies on significantly less and simpler input.

Table V 1 shows that the algorithms performed roughly well on the analyzed traces. On the other hand, the fact that all three methods use heuristics implies that some fine tuning work may be needed to fit the methods to other traces or new applications.

For comparison purposes, the results from a byte signature based analysis (referred to as DPI) are also shown in Table V 1. This analysis was made possible by the manual identification of flows for comparison. When comparing the payload analysis with other identification techniques, it becomes clear that these techniques achieve slightly better results than those shown by the DPI.

First, these techniques can capture the behavior of an application, sometimes finding new applications that still had no payload signature but behaved similarly to applications of the same type. Second, this comparison is unfair, since most papers with behavioral analysis only considered TCP flows with well-behaved sessions (started and finished during the trace), while the DPI was performed in a real, non-normalized trace.

In addition to that, the BLINC and the On the Fly analyses used DPI as a baseline for identification, and therefore their results must be interpreted as relative to DPI, not a comparison with actual applications. Finally, the Bayesian and the On the Fly analysis lack the completeness metric, and therefore their accuracy must not be taken for granted or as absolute. This is because, for an independent identification technique (not combined with other technique), a high accuracy with a low completeness is just as good as a low accuracy with a good completeness. Since the completeness is unknown, the meaningfulness of their accuracy also becomes unknown.

One way of getting around the uncertainty provided by the heuristics is to run more algorithms in parallel, compare their results, and conclude the final application classification decision based on the result of the comparison, as introduced by [63] (with their own algorithms). This approach also has the advantage that mismatching classification results are recognized automatically. Furthermore, such traffic may be dumped separately for further analysis and the knowledge gained can be incorporated into the algorithms.

Authors of [63] executed different classification methods on different measurements one-by-one and then the results were combined by their suggested decision mechanism. The accuracy and the completeness of the classification methods on different application types compared to their combined classification method can be seen in [63] (Table 1 of the paper). Some results for traffic identification mechanisms are presented next.

The joint application – and the comparison run on the same input data – in [63] shows that some methods are stronger in accuracy and others provide more complete results (see Table I of [63]). As a consequence, the application classification decision is a trade-off between the amount of traffic left unknown and the bigger likelihood of erroneous classification.

VI. Conclusions

Many techniques for network management and application identification do exist, but some suffer from legal problems (signature-based packet payload analysis) while others (inference-based) only identify a few applications correctly. Well-known-ports are no longer an answer, since many applications, especially those with a high network volume (e.g., P2P file sharing), bypass the rules and use known ports of other services. Payload-based schemes are very time-consuming, therefore should not be utilized in real-time in high-speed links, except when using high cost specialized hardware in specific network links (up to 1Gbps). Flow-based schemes (inference) lose information, and even these require sampling on very high speed links, depending on the routers. Some authors claim their inference-based methods achieve

Table I
ACCURACY AND COMPLETENESS OF THE IDENTIFICATION METHOD

Application	Metric	BLINC	Bayesian	On the Fly	Payload Analysis
WWW	Completeness	69-97%	-	-	134%
	Accuracy	98-100%	99.27	-	91%
-HTTP	Accuracy	-	-	99%	-
-HTTPS	Accuracy	-	-	81.8%	-
Mail	Completeness	78-95%	-	-	78%
	Accuracy	85-99%	94.78%	-	97%
-SMTP	Accuracy	-	-	84.4%	-
-POP3	Accuracy	-	-	0%	-
-POP3S	Accuracy	-	-	89.8%	-
Bulk/File Transfer (FTP)	Completeness	95%	-	-	26%
	Accuracy	98%	82.25%	87%	99%
NNTP	Accuracy	-	-	99.6%	-
Chat	Completeness	68%	-	-	76%
	Accuracy	98%	-	-	97%
Network Management/System	Completeness	85-97%	-	-	75%
	Accuracy	88-100%	-	-	95%
Services (Server)	Accuracy	-	63.68%	-	-
Database	Accuracy	-	86.91%	-	-
Multimedia / Streaming	Completeness	-	-	-	3%
	Accuracy	-	80.75%	-	98%
Peer-to-Peer (File Sharing)	Completeness	84-90%	-	-	61%
	Accuracy	96-98%	36.45%	-	99%
-Kazaa	Accuracy	-	-	95.24%	-
-Edonkey	Accuracy	-	-	84.2%	-
SSH	Accuracy	-	-	96.92%	-
Tunneled	Completeness	-	-	-	120%
	Accuracy	-	-	-	10%

high efficiency and precision, but it greatly varies with the traffic pattern studied.

Finally, there is no final answer for application recognition in IP networks, especially one good enough to be used in traffic shaping (i.e., different applications get different service from the network), filtering (i.e., threats/attacks can be blocked) and billing (i.e., per-application charging rates). The next sections present some recommendations for traffic classification and list some open research questions.

A. Requirements for Application Classification Analysis

Despite the importance of traffic classification, an accurate method that can reliably address this problem is still to be found. As far as existing traffic identification platforms offered by vendors (hardware and software) are concerned, the use of well known ports is still the main technique, along with packet payload recognition: an analysis of the headers or payload of packets is used to identify traffic associated with a particular port and thus of a particular application. However, both methods have some pitfalls. Well-known port numbers can no longer be used to reliably identify network applications. There is a variety of new Internet applications that either do not use well-known port numbers or use other protocols, such as HTTP, as wrappers in order to go through firewalls without being blocked. One consequence of this is that a simple inspection of the port numbers used by flows eventually leads to an inaccurate classification of network traffic.

As for payload analysis, it suffers from poor scalability, is unable to classify encrypted traffic and has legal problems in some countries. In other words, taking into account empirical application trends and the increasing use of encryption, the traffic classification issue should consider the following constraints:

- No access to user packet payload, except when possible/authorized/legal and even though only on special cases, to discover information that may not be gathered or inferred from other means;
- Well-known port numbers cannot identify applications reliably; and
- Only information that current flow collectors (e.g. NetFlow) provide can be used, except when a powerful infrastructure for packet capture is provided.

B. Open Research Questions

Based on this survey of traffic identification papers, the authors identified some issues that still remain open, and are discussed next.

- At first, the best level of detail for measurements is still not defined. This leads to a multidimensional problem constrained by existing equipment for measurements and main traffic characteristics. From the research point of view, the problem is to find the minimal amount of data that needs to be measured in order to classify applications. However, storing the minimal amount of data may not be the best solution, since additional data may be needed to validate results. Furthermore, in practice, measured results usually raise additional questions and existing extra measurement may help in prompt replies. In any case, the pros and cons of different measurement levels are not thoroughly understood. On the other hand, networks carry high traffic volume, which imposes a higher burden on traffic measurements. One way to deal with this problem is to apply sampling or other filtering techniques, e.g. measure the traffic of selected subscribers. It is not clear how much sampling can be used to keep a certain level of accuracy. It is also not clear how much information is lost given a certain sampling approach.

• Since many applications refuse to use well-known ports, mostly for bypassing firewalls, identifying applications out of flow-records is still an art. In other words, as shown in sections IV and V, most existing proposals are based on inference techniques that do not yield always acceptable results, so that a great deal of fine-tuning for specific applications and traces is still needed Therefore, the question of how to create an approach able to achieve acceptable levels of accuracy and completeness for a variety of different applications, networks and time periods is still an open question.

• All application classification methods leave parts of the traffic unclassified. For example, in the BLINC [43] use-case, it could not classify 1-7% of the flows and 6-17% of the bytes, depending on the trace. There are a number of reasons for not being able to classify a flow: new application, ciphered traffic or misbehaved application. Therefore, a general goal is to reduce the amount of unclassified traffic.

• Application classification methods use heuristics. As a consequence of that, their validation will always be problematic, since another, reliable, method will be required for this validation. Some heuristics are more reliable than others, e.g. a long byte signature is more accurate than a short one. Application classification algorithms should formalize and describe the reliability of the classification decision for each flow or packet (e.g., marking traffic for which the classification is certain).

• Also, updating an identification technique is time consuming. In addition to the improvement of the technique (increase of accuracy), this motivates the research of methods that can recognize new protocols or changes in protocol versions of the same application type automatically. Although new applications sometimes impose changes in the network traffic patterns (e.g., Voice over IP usage, Video on Demand, Internet Worms), presently network operators cannot always promptly react to them. The automatic recognition of new application types may also be a problem for research, but it is controversial if this can be solved at all. Therefore, the question is if there is a way to automatically profile new applications or alternatively they should be penalized (e.g. by shaping) until they are thoroughly understood.

• While many commodity programs already exist for network management, they do not show explicitly what the network manager should know about what is happening to the network. They provide some metrics and the network manager must know already how to interpret the results. Are there any metrics for understanding the behavior of users? Are there any metrics that are specific for broadband or home users?

• Since broadband users tend to stay connected for a longer time period and sometimes even let their network access active just for connectivity (e.g., over-night downloads, VoIP applications waiting for a call), it is important to know the current typical traffic profile for them. Is there a traffic profile trend towards the greater usage of certain applications? How do these applications behave and how will they behave with the growing number of users?

• Choosing the right place where to make the measurements in the network may influence the results. For example, an inspection made at the access networks may show the presence of DoS user traffic whereas a probe at the domain level may not show this as it is usually filtered before hand by corporate firewalls.

All of these questions remain open and any answers would greatly improve the way network managers operate their networks and adapt to changes.

C. Final Remarks

Internet measurement is a very dynamic and wide field; all the time new approaches to network management, application profiling and traffic modeling are proposed, each analyzing a different aspect.

Packet-based application inference has some issues that may not be circumvented technologically. Flow-based application inference is still an incipient field of study, despite the many papers on the subject. Using present day research, none of them achieve a high accuracy with a high precision in a broad range of applications. Further study is required on a new technique for dependable application detection.

REFERENCES

[1] Azzouna, Nadia Ben and Guillemin, Fabrice, *Analysis of ADSL Traffic on an IP Backbone Link*, IEEE IEEE Global Telecommunications Conference 2003, San Francisco, USA, December 2003.

[2] Sullivan, Mark, *Surveys: Internet Traffic Touched by Youtube*, Light Reading, http://www.lightreading.com/document.asp?doc_id=115816, January 2007.

[3] Crovella, Mark and Krishnamurty, Balachander, *Internet Measurement: Infrastructure, Traffic and Applications*, book, ISBN-13 978-0470014615, Wiley, 2006.

[4] Zhang, Jian and Moore, A., *Traffic Trace Artifacts due to Monitoring Via Port Mirroring*, in: Proc. 5th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services, 2007 (E2EMON '07), May 2007.

[5] Cho, Kenjiro; Fukuda, Kenshue; Esaki, Hiroshi and Kato, Akira, *The Impact and Implications of the Growth in Residential User-to-User Traffic*, ACM SIGCOMM 2006, Pisa, Italy, September 2006.

[6] Marques Nt., H. T.; Rocha, L. C., Guerra, P. H., Almeida, J. M., Meira, W., and Almeida, V. A., *Characterizing Broadband User Behavior*, Proc. 2004 ACM Workshop on Next-Generation Residential Broadband Challenges, NRBC '04. ACM Press, New York, NY, 11-18, October 2004.

[7] Balachandran, Anand; Voelker, Geoffrey M.; Bahl, Paramvir and Ragan, P. Venkat, *Characterizing user behavior and network performance in a public wireless LAN*, Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 195-205, 2002.

[8] Karagiannis, Thomas; Broido, Andre; Brownlee, Nevil; Claffy, K.C. and Faloutsos, Michalis, *Is P2P dying or just hiding?*, IEEE Global Telecommunications Conference, November 2004.

[9] Karagiannis, Thomas; Broido, Andre; Faloutsos, Michalis and Claffy, K.C., *Transport Layer Identification of P2P Traffic*, Internet Measurement Conference (IMC '2004), October 2004.

[10] Moore, Andrew W. and Papagiannaki, Konstantina, *Toward the Accurate Identification of Network Applications*, Passive and Active Measurement Workshop (PAM 2005), March 2005.

[11] Madhukar, A. and Williamson, C., *A Longitudinal Study of P2P Traffic Classification*, 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, September, 2006.

[12] Cherry, S., *The VoIP Backlash*, IEEE Spectr., October 2005, http://spectrum.ieee.org/oct05/1846.

[13] Ohm, Paul; Sicker, Douglas and Grunwald, Dirk, *Legal Issues Surrounding Monitoring During Network Research*, (invited paper), Proc. 7th ACM SIGCOMM Conference on Internet Measurement, San Diego, USA, 2007.

[14] Kumar, Anurag; Manjunath, D. and Kuri, Joy, *Communication Networking – an Analytical Approach*, Morgan Kaufmann Publishers, Elsevier Inc., pp. 656-664, 2004.

[15] Shah, Niraj, *Understanding network processors*, Master's thesis, Department of Eletrical Engineering and Computer Science, University of California, Berkley, 2001.

[16] Orebaugh, A., Morris, G., Warnicke, E. and Ramirez, G., *Ethereal Packet Sniffing*, Syngress Publishing, February 2004.

[17] Cisco IOS NetFlow, *Introduction to Cisco IOS NetFlow - A Technical Overview*, White Paper, Last updated: October 2007, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html.

[18] Sadasivan, G.; Brownlee, N.; Claise, B. and Quittek, J., *Architecture for IP Flow Information Export*, IP Flow Information Export WG (expires in March 2007), September 2006.

[19] Iannaccone, G., *Fast Prototyping of Network Data Mining Applications*, 7th Passive and Active Measurement Workshop. Adelaide, Australia, March 2006.

[20] Duffield, N. and Lund, C., *Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure*, ACM Internet Measurement Conference 2003.

[21] Duffield, N., Lund, C. and Thorup, M., *Estimating Flow Distributions from Sampled Flow Statistics*, ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.

[22] Duffield, N.; Lund, C. and Thorup, M., *Learn more, sample less: control of volume and variance in network measurement*, IEEE Trans. Inform. Theory, vol. 51, no. 5, pp. 1756-1775, 2005.

[23] Fernandes, S.; Kamienski, C.; Kelner, J.; Sousa, D.; and Sadok, D., *A Stratified Traffic Sampling Methodology for Seeing the Big Picture*, The International Journal of Computer and Telecommunications Networking (Elsevier Computer Networks), 2008.

[24] Sommer, R. and Feldmann, A., *NetFlow: information loss or win?*, In Proc. 2nd ACM SIGCOMM Workshop on internet Measurement (Marseille, France, November 06 - 08, 2002). IMW '02. ACM Press, New York, NY, 173-174.

[25] Liu, Yong; Towsley, Don; Ye, Tao and Bolot, Jean, *An Information-theoretic Approach to Network Monitoring and Measurement*, Internet Measurement Conference, IMC '05, 2005.

[26] Liu, W.; and Boutaba, R., *pMeasure: A Peer-to-Peer Measurement Infrastructure For the Internet*, Computer Communications Journal, Special Issue on Monitoring and Measurements of IP Networks, 2005.

[27] Paxson, Vern, *Bro: A system for detecting network intruders in real-time*, Computer Networks, vol. 31, no. 23-24, pp. 2435-2463, 1999.

[28] Wallerich, J., Dreger, H., Feldmann, A., Krishnamurthy, B. and Willinger, W., *A methodology for studying persistency aspects of internet flows*, SIGCOMM Comput. Commun. Rev. 35, 2 (April 2005), 23-36.

[29] Xu, Kuai; Zhang, Zhi-Li and Bhattacharya, Supratik, *Profiling Internet Backbone Traffic: Behavior Models and Applications*, ACM SIGCOMM 2005.

[30] Lan, K. and Heidemann, J., *A measurement study of correlations of internet flow characteristics*, Computer Networks 50, 1 (Jan. 2006), 46-62.

[31] Papagiannaki, K., Taft, N., Bhattacharyya, S., Thiran, P., Salamatian, K. and Diot, C., *A pragmatic definition of elephants in internet backbone traffic*, In Proc. 2nd ACM SIGCOMM Workshop on internet Measurement (Marseille, France, November 06 - 08, 2002). IMW '02. ACM Press, New York, NY, 175-176.

[32] Brownlee, N. and Claffy, K.C., *Understanding Internet traffic streams: dragonflies and tortoises*, Communications Magazine, IEEE, vol.40, no.10, October 2002, pp. 110-117.

[33] Park, Kihong & Willinger, Walter, *Self-Similar Network Traffic and Performance Evaluation*, Wiley-Interscience, ISBN: 978-0471319740, First Edition, January 2000.

[34] Karagiannis, T.; Molle, M. and Faloutsos, M., *Long-range dependence – Ten years of Internet traffic modeling*, IEEE Internet Computing magazine, volume 8, number 5, pp. 57-64, September 2004.

[35] Sarvotham, S., Riedi, R., and Baraniuk, R., *Connection-level analysis and modeling of network traffic*, In Proc. 1st ACM SIGCOMM Workshop on Internet Measurement (San Francisco, California, USA, November 01 - 02, 2001).

[36] Crovella, Mark, *Network Traffic Modeling*, PhD lecture, Aalborg University, February 2004.

[37] Chen, Thomas M., *Network Traffic Modeling*, Chapter in "The Handbook of Computer Networks", Bigdoli, Hossein, editor, volume III, Wiley, 2007.

[38] Kim, Myung-Sup; Won, Young J. and Hong, James W., *Characteristic analysis of internet traffic from the perspective of flows*, Computer Communications Journal, 2005.

[39] Brownlee, Nevil, *Some Observations of Internet Stream Lifetimes*, Lecture Notes in Computer Science, Volume 3431, January 2005, pp. 265 – 277.

[40] Willinger, Walter; Taqqu, Murad S.; Sherman, Robert and Wilson, Daniel V., *Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level*, IEEE/ACM Trans. Networking, volume 5, pp. 71-86. February 1997.

[41] Bacelar, Rodrigo; Kamienski, C. A.; Callado, Arthur; Fernandes, Stênio; Mariz, Dênio and Sadok, Djamel, *Performance evaluation of P2P VoIP applications*, Proc. 17th International workshop on Network and Operating Systems Support for Digital Audio & Video (NOSSDAV 2007), Urbana-Champaign, USA. 2007.

[42] Sinha, A.; Mitchell, K. and Medhi D., *Flow-Level Upstream Traffic Behavior in Broadband Access Networks: DSL versus Broadband Fixed Wireless*, IEEE IPOM, 2003.

[43] Karagiannis, T., Papagiannaki, K. and Faloutsos, M., *BLINC: Multilevel Traffic Classification in the Dark*, ACM SIGCOMM 2005, August/September 2005.

[44] Moore, Andrew and Zuev, Denis, *Internet traffic Classification Using Bayesian Analysis Techniques*, Proc. 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, ACM Press, 2005, pp. 50-60.

[45] IANA, *Port Numbers*, http://www.iana.org/assignments/port-numbers.

[46] Sen, Subhabrata; Spatscheck, Oliver and Wang, Dongmei, *Accurate, Scalable InNetwork Identification of P2P Traffic Using Application Signatures*, Proc. 13th international conference on World Wide Web, 2004, pp. 512-521.

[47] Haffner, P.; Sen, S.; Spatscheck, O. and Wang, Do., *ACAS: Automated Construction of Application Signatures*, MineNet 2005.

[48] Kim, H. A. and Karp, B., *Autograph: Toward Automated, Distributed Worm Signature Detection*, Security 2004.

[49] Li, Z.; Sanghi, M.; Chen, Y.; Kao, M.-Y. and Chavez, B., *Hamsa: Fast Signature Generation for Zero-day Polymorphic Worms with Provable Attack Resilience*, IEEE Symposium on Security and Privacy 2006.

[50] Markatos, E. P.; Antonatos, S.; Polychronakis, M. and Anagnostakis, K. G., *Exclusion-based Signature Matching for Intrusion Detection*, In Proc. IASTED International Conference on Communications and Computer Networks, 2002.

[51] Erdogan, O. and Cao, P., *Hash-AV: Fast Virus Signature Scanning by Cache-Resident Filters*, IEEE Global Telecommunications Conference, St. Louis, USA, 2005.

[52] Roughan, Matthew; Sen, Subhabrata; Spatscheck, Oliver and Duffield, Nick, *Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification*, Proc. 4th ACM SIGCOMM Conference on Internet Measurement, Taormina, Italy, 2004.

[53] Zander, S.; Nguyen, T. and Armitage, G., *Automated Traffic Classification and Application Identification Using Machine Learning*, IEEE Conference on Local Computer Networks, 2005.

[54] McGregor, A.; Hall, M.; Lorier, P. and Brunskill, A., *Flow Clustering Using Machine Learning Techniques*, Passive and Active Network Measurement Workshop, 2004.

[55] Erman, Jeffrey; Mahanti, Anirben and Arlitt, Martin, *Internet Traffic Identification using Machine Learning*, Proceeding of the 49th IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, USA, November 2006.

[56] Williams, Nigel; Zander, Sebastian and Armitage, Grenville, *A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification*, ACM SIGCOMM Computer Communication Review, volume 36, issue 5, October 2006.

[57] Bonfiglio, Dario; Mellia, Marco; Meo, Michela; Rossi, Dario and Tofanelli, Paolo, *Revealing Skype Traffic: when randomness plays with you*, ACM SIGCOMM 2007 Data Communication Festival (SIGCOMM 2007), Tokyo, Japan, August 2007.

[58] Suh, Kyoungwon; Figueiredo, Daniel R., Kurose, Jim and Towsley, Don, *Characterizing and Detecting Skype-Related Traffic*, The 25th Conference on Computer Communications. Barcelona, Spain, April 2006.

[59] Chhabra, Parminder; John, Ajita and Saran, Huzur, *PISA: Automatic Extraction of Traffic Signatures*, 4th International IFIP-TC6 Networking Conference, May 2005.

[60] Bernaille, Laurent; Teixeira, Renata and Salamantian, Kavé, *Early Application Identification*, Second Conference on Future Networking Technologies, December 2006.

[61] Bernaille, Laurent; Teixeira, Renata; Akodkenou, Ismael; Soule, Augustin and Salamantian, Kave, *Traffic Classification On The Fly*, ACM SIGCOMM Computer Communication Review, Volume 36, Number 2, April 2006, pp. 23-26.

[62] Bernaille, Laurent and Teixeira, Renata, *Early Recognition of Encrypted Applications*, in Proc. 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, April 2007.

[63] Szabó, G.; Szabo, I. and Orincsay, D., *Accurate Traffic Classification*, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 2007.

[64] Silvestre, G.; Fernandes, S.; Kamienski, C.; Sousa, D. and Sadok, D., *Padrões de Comportamento de Tráfego P2P*, Proc. III Workshop of Peer-to-Peer (WP2P 2007), June 2007.
[65] Andersson, L.; Davies, E. and Zahng, L., *Report from the IAB workshop on Unwanted Traffic*, Internet-Draft, IETF Network Working Group, February 2007.
[66] Soule, A.; Salamantian, K. and Taft, N., *Combining Filtering and Statistical Methods for Anomaly Detection*, Proc. Internet Measurement Conference 2005, pp. 331-344, 2005.
[67] Huang, C.-T.; Thareja, S. and Shin, Y.-J., *Wavelet-based Real Time Detection of Network Traffic Anomalies*, Proc. Workshop on Enterprise Network Security (WENS 2006), August 2006.
[68] Magnaghi, A.; Hamada, T. and Katsuyama, T., *A Wavelet-based Framework for Proactive Detection of Network Misconfigurations*, Proc. ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality, pp. 253-258, 2004.
[69] Lakhina, Anukool; Crovella, Mark and Diot, Christophe, *Mining Anomalies Using Traffic Feature Distributions*, ACM SIGCOMM Computer Communication Review, volume 35, Issue 4, pp. 217-228, 2005.
[70] Szabó, G.; Orincsay, D.; Malomsoky, S. and Szabó, I., *On the Validation of Traffic Classification Algorithms*, Passive and Active Measurements Workshop, April 2008.
[71] Jiang, Hongbo; Moore, Andrew; Ge, Zihui; Jin, Shudong and Wang, Jia, *Lightweight Application Classification for Network Management*, SIGCOMM Workshop on Internet Network Management (INM), August, 2007.
[72] Karagiannis, T.; Papagiannaki, K.; Taft, N. and Faloutsos, M., *Profiling the End Host*, in Proceedings of the 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, pp. 186-196, April 2007.
[73] Kundu, Sumantra; Pal, Sourav; Basu, Kalyan and Das, Sajal, *Fast Classification and Estimation of Internet Traffic Flows*, in Proc. 8th International Conference on Passive and Active Measurement, Louvain-la-Neuve, Belgium, April 2007.
[74] Tai, Masaki; Ata, Shingo and Oka, Ikuo, *Fast, Accurate, and Lightweight Real-time Traffic Identification Method based on Flow Statistics*, in Proc. 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, pp. 255 -299, April 2007.

Arthur de C. Callado received a B.Sc. degree in Computer Science in 2000 from the Federal University of Ceará in Fortaleza, Brazil and the M.Sc. and Ph.D. degrees in Computer Science in 2004 and 2009, respectively, from the Federal University of Pernambuco, in Recife, Brazil.

The author's main interests include Internet measurement, monitoring, quality of service and VoIP. He is a fellow researcher in the Network and Telecommunications Research Group from the Federal University of Pernambuco.

Mr. Callado is a member of the Brazilian Computer Society.

Carlos A. Kamienski (M'2007) became a member of IEEE in 2007. He received his Ph.D. in computer science from the Federal University of Pernambuco (Recife PE, Brazil) in 2003. His current research interests include policy-based management, traffic measurement and analysis and ambient networks. He is currently an associate professor of computer networks at the Federal University of the ABC in Santo André SP, Brazil. He has been involved in several research projects funded by Brazilian agencies and also in partnership with telecom companies.

Géza Szabó received the M.Sc. degree in Computer Science in 2006 from the Budapest University of Technology and Economics, in Budapest, Hungary. The author's main interests include internet traffic classification and modeling.

He works for Traffic Analysis and Network Performance Laboratory of Ericsson Research in Budapest, Hungary.

Balázs Péter Gerő received the M.Sc. degree in computer science from the Technical University of Budapest in 1998. His research interests include performance and traffic modeling of mobile networks and restoration methods of transport networks.

He is currently with Ericsson Traffic Analysis and Network Performance Laboratory, in Budapest, Hungary.

Judith Kelner received her Ph.D. degree from the Computing Laboratory, the University of Kent at Canterbury in 1993.

She is currently a Professor at the Computer Science Center in the Federal University of Pernambuco (Brazil) with research interests in the areas of of Multimedia, Virtual Reality systems and Computer Networks. Professor Kelner is Senior project leader at the GPRT research group at the Federal University of Pernambuco.

Stênio F. de L. Fernandes (M'1997) became a member of IEEE in 1997. He received a B.S. and a M.S. degree in Electronic Engineering from the Federal University of Paraiba (UFPB, now UFCG) in 1992 and 1996, respectively. He also received a Ph.D. in Computer Science from the Federal University of Pernambuco (UFPE) in 2006. The author's main interests include Internet traffic measurement, modeling and analysis, systems performance evaluation, Internet congestion control, multimedia streaming in the Internet with VoIP and P2PVideo and virtual worlds.

He holds a faculty position at the Federal Institute for Education, Science and Technology of Alagoas (IF-AL) and is also a senior researcher at the Network and Telecommunications Research Group of the Federal University of Pernambuco.

Djamel F. H. Sadok (M'95-SM'03) became a member of IEEE in 1995 and became a Senior Member in 2003. He received his Ph.D. degree from Kent University in 1990. His current research interests include traffic engineering of IP networks, wireless communications, broadband access, and network management. He currently leads a number of research projects with many telecommunication companies.

He is currently a professor of computer networks at the Computer Science Department of the Federal University of Pernambuco, Recife PE, Brazil. He is one of the cofounders of GPRT, a research group in the areas of computer networks and telecommunications. From 1990 to 1992 he was a research fellow in the Computer Science Department, University College London.

Dr. Sadok is a member of the editorial body of the Journal of Networks and of the Reviewer for IEEE Communications Magazines.