A Survey on Internet Traffic Identification and Classification

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

***Abstract*— The area of Internet traffic measurement has advanced enormously in the last couple of years. This advancement is mostly based on the enormous growth in the number of users connected, in the increase in user access speeds and in the appearance of network-hungry applications. These changes greatly affected the work of Internet Service Providers and network administrators, which have to deal with increasing network users and capacity demands and abrupt traffic changes caused by new applications. This survey explains the main problems 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 of each approach. Second, this work cites the techniques for traffic analysis available in the literature, along with the analyses performed by the authors. These techniques include signature-matching, sampling and inference. Third, this work shows the trends in application behavior analyses and cites important and recent references in the subject. Lastly, this survey enlists the open topics of research by explaining the questions that are still open in traffic analysis and application detection and makes some final remarks.**

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

I. Introduction

CHARACTERIZATION of Internet traffic has become over the past few years one of the major challenging issues in telecommunication networks [2]. 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 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 dial-up user.

In general, characterization of Internet traffic provides input 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 point 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 analyses (still without any papers published) [3], last year's Video sharing traffic has greatly increased Internet usage, surpassing P2P. However, conducting a sound Internet measurement study is a difficult undertaking [4]. Previous investigations suffer from known limitations, such as limited measurement duration or coverage, loss of information during the measurement process and failure to identify the application correctly.

Additionally, the availability of broadband connections for Internet users is continually growing; particularly those based on cable TV and ADSL infrastructure and technologies. Such availability opened up new ways of resource usage for both home users and small organizations. Since the broadband Internet connection is always available as well as it has increased its 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 peer-to-peer (P2P) systems for resource sharing, particularly audio and video files. In other words, the increased capacity and availability has led to a complex behavior for a typical user [5], very different from 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 this more complex behavior. Furthermore, the current trend of moving phone calls from the PSTN to the

Manuscript received September 7, 2007. (Write the date on which you submitted your paper for review.) This work was supported in part by Ericsson Brazil under Grant UFP.17.

A. de C. Callado is with the Federal University of Pernambuco in Recife, PE 50670-901 Brazil (corresponding author, phone: +55 81 2126-8954; fax: +55 81 2126-8955; e-mail: arthur@gprt.ufpe.br).

C. A. Kamienski is with the Federal University of ABC in Santo André, SP 09.210-170 Brazil (e-mail: cak@gprt.ufpe.br).

Stênio Fernandes is with the Federal Center for Education in Technology in Maceió, AL 57020-510 Brazil (e-mail: stenio.fernandes@ieee.org).

D. F. H. Sadok is with the Federal University of Pernambuco in Recife, PE 50670-901 Brazil (e-mail: jamel@gprt.ufpe.br).

G. Szabó is with the Traffic Analysis and Network Performance Laboratory of Ericsson Research in Budapest, Hungary (e-mail: geza.szabo@ericsson.com).

B. P. Gerö is with the Traffic Analysis and Network Performance Laboratory of Ericsson Research in Budapest, Hungary (e-mail: balazs.peter.gero@ericsson.com).

Internet via VoIP applications represents a threat to telephony companies and its effects have not yet been completely understood.

Chapter 2 compares packet-based and flow-based analyses techniques. Chapter 3 explains some techniques used for traffic analyses. Chapter 4 focuses application behavior analyses. Chapter 5 compares some 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 for research in traffic measurement.

II. Packet-Based and Flow-Based Measurement Techniques

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 peer-to-peer) 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 residential users 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, DoS attacks, routing problems, etc.), 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 even for usage-based pricing. 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). Traffic identification, however, is inherent to traffic classification, since one may not classify before identification.

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. On the other hand, they cannot reveal network characteristics influenced by users, due to the fact that active measurements send user behavior independent packets. 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. Most network operators are unwilling to generate extra traffic for active measurement when there is a passive alternative. Therefore, most measurement techniques of Internet traffic falls into the area known as passive measurement, i.e., they do not send packet probes in order to obtain the envisioned traffic metrics.

Passive techniques are carried out by observing network traffic packets and flows. When observing packets, most techniques consist of capturing packet headers and analyzing them. There is an on-going discussion about some legal issues when inspecting packet payload [7], i.e. packet payload inspection is forbidden in some countries. On the other hand, flow-based measurement deals with a summary of unidirectional streams of packets passing through a given router. For an in-depth characterization of network traffic, passive measurements can be undertaken in two levels, namely packet-based and flow-based measurements. Another type of source for traffic analysis is the Management Information Base (MIB) data, available through the Simple Network Management Protocol (SNMP), which is 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 about packets (except the total number of packets seen at a given interface, which can be polled for a low-resolution total volume analysis) or flows, and consequently some vital information of passing communications are 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. It is appropriate for total volume measurement and per-interface traffic accounting.

Passive measurement techniques are particularly suitable for traffic engineering and capacity planning because they show traffic dynamics and distribution. The main problem with passive measurement is dealing with a massive amount of data, since it scales with link capacity. In other words, the volume of captured data can become very large on high-capacity links or when dealing with a massive amount of users.

A. *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 interesting 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 famous tool that allows one to look closer at network packets and make

[1] http://www.tcpdump.org

some statistical analysis out of the trace files. Ethereal[2] [8] or Wireshark[3], which adds a user-friendly GUI to tcpdump and includes many traffic signatures, 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 attacks. A set of other packet-related tools may be found in the Internet[5].

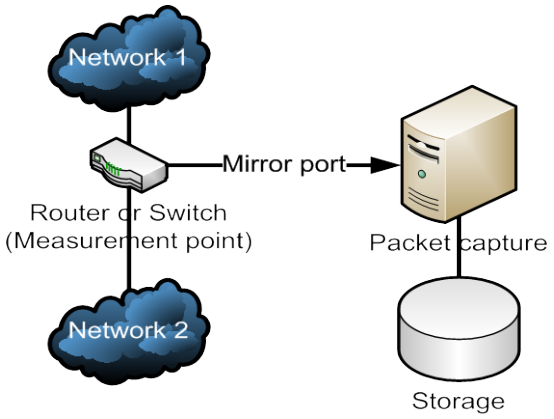

Fig. 1: Packet measurement topology.

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 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 traffic mirroring for another machine (Fig. 1, with a router), which may alleviate processing usage. In the example shown in Fig. 1, in the process of capturing packets a machine is directly connected to the measured router (or switch), and the router (or switch) must support 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. Then, 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 prohibitive. 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., low-end network interface, low CPU power) will not be able to capture packets at full wirespeed, and may miss some packets.

B. Flow-Level Passive Measurements

At a macroscopic level, measurements are performed on flows. In this case, aggregation rules are necessary to match packets into flows. Examples of collected data are the number of flows per unit of time, flow bitrate, flow size and flow duration. Examples of tools that deal with flows are Cisco NetFlow[6] [9] (the de facto standard) and Juniper JFlow[7].

Cisco was the first to propose and implement flow-level captures. NetFlow provides a set of services for IP applications, including network traffic accounting, usage-based network billing, network planning, security, Denial of Service (DoS) monitoring capabilities, and network monitoring. Currently, the most important technology for measuring and exporting traffic flows is Cisco's NetFlow[8].

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

The specification of the 48–byte long NetFlow record is detailed in Table I.

TABLE I EXPORT PACKET FLOW RECORD FORMAT

Field	Description	Bytes
Source IP	Flow Source IP Address	4
Destination IP	Flow Destination IP Address	4
Next Hop IP	The IP address of the next router in the flows' path	4
Inbound Interface ID	SNMP index of the flows' incoming interface	2
Outbound Interface ID	SNMP index of the flows' outgoing interface	2
Packet Count	Number of packets seen in the flow	4
Byte Count	Number of bytes seen in the flow	4
Start Time	Value (milliseconds) of SysUpTime when first packet was seen	4
End Time	Value (milliseconds) of SysUpTime when last packet was seen	4
Source Port	Source TCP or UDP port	2
Destination Port	Destination TCP or UDP port	2
Pad Byte	Ignored byte	1
TCP Flags	All TCP flags' bits ever used in the flow	1

[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

Layer 4 Protocol	Transport protocol (e.g.: TCP, UDP, ICMP)	1
Type of Service	Type of Service field of the IP header used in the flow	1
Source AS ID	Origin Autonomous System (network/backbone) identification number	2
Destination AS ID	Destination Autonomous System (network/backbone) identification number	2
Source Mask Bits	Number of bits in source network mask	1
Destination Mask Bits	Number of bits in destination network mask	1
Pad Bytes	Ignored bytes	2

Although NetFlow v5 provides many fields of information, in practice many programs fail to correctly fill all the fields, so 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. The 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 flows.

JFlow also provides a similar set of functionalities and supports NetFlow export formats. In fact, most software developers of flow collectors along with the leading companies in the router-related industry are heading to follow the standardization of flow records from the IETF standard proposal IPFIX [10], which was previously a Cisco proposal (called NetFlow 10). There are good software tools for working with Netflow and JFlow flow traces, such as Flow-Tools[11] and Ntop[12]. Ntop works also as a sniffer, i.e., it has packet capture features.

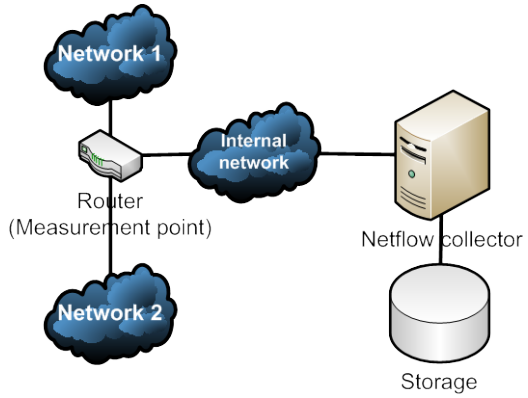

Fig. 2: Flow Measurement Topology.

In Fig. 2, we see the topology of a Netflow measurement. Here, the flow collector does not have to be directly connected to the router where the measurement is performed – it 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 traffic analysis.

C. *Sampling Techniques*

Both flow monitoring tools and packet level capture tools suffer from 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 is a crucial technique for a scalable Internet monitoring, which is a reasonable statistical strategy for dealing with high volume 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 at the same time as reducing the volume of collected data. Indeed, such reduction makes the monitoring process scalable for a variety of network sizes. In order to transform the obtained partial data into information or knowledge about the 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) [11][12] and flow selection (for statistical analysis of the original traffic) [13]. Sampling may be applied during the capture (packet level, as done on CISCO routers[13]) or after data classification and aggregation (flow level). Please note that sampling flow measurement has become increasingly employed at packet level in order to reduce the computational overhead 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 [13]. 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 the same problem; the average traffic volume estimated with random sampling is biased towards a smaller average value. It means that any traffic monitoring technique based on uniform or random

[10] http://manageengine.adventnet.com/products/netflow/help/installation/setup-cisco-netflow.html
[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

sampling will provide an underestimation of traffic volume.

It is advocated [14] that reducing the number of collected traffic flows (regardless 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. traffic flow size and duration, sampling each subpopulation (stratum) independently may be advantageous for reducing the sampling error. In [14], 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 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 our 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 on both routers and collectors. It may also have an 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 processing overhead. For example, in the research work presented in [15], the authors perform a study on the accuracy and overhead of NetFlow's sampling feature in a Cisco GSR 12000 series router. In their experiments, they turned 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 the bandwidth usage due to export packets grows linearly as the number of flows increase. Another result is 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 alluded to earlier in this report, 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 management. Essentially, there is a broad avenue for future research in this field, since defining strategies – beyond sampling - for dealing with the huge amount of network traffic data is an overwhelming task. In addition, obtaining an in-depth knowledge of the trade-off related to the granularity of traffic measurement in a major ISP is an open topic.

In [16], 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 compress 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 [17] Liu and Boutaba describe a peer-to-peer 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 the tasks. All experiments and simulation results point out that p2p-based measurement system are very promising, but no results of a practical measurement are shown.

Due to the strong restrictions on packet-level analysis, we envision that most network monitoring and analysis systems will be based on flow-based techniques. Although the Internet scientific community has to overcome limitations on this approach, we observed that important efforts have been made towards gathering sound information from flow records.

As an important step for a cautious deployment of any new technique for network management, the research work in [15] analyses the loss of information caused by TCP 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 breaking each 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 [18]. When analyzing the differences between flow connections and packet connections, FLOW-REDUCE heuristic makes a good match for long-lived connections, more specifically, those longer than 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.

A. Classification: Volume, Duration, Rate and Burstiness

In a similar approach to [15], the research work in [19] performs a multi-scale and multi-protocol analysis to explore the persistency 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, MIB/SNMP). 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 by 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 (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 [20], Lan and Heidemann [21] 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 by size (e.g., elephant or heavy-hitter and mouse), by duration (e.g., tortoise and dragonfly), by rate (e.g., cheetah and snail) and by burstiness (e.g., porcupine and stingray, but also called alpha and beta traffic). Specifically, a flow in a trace is 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 a Tortoise if its duration is longer than the average duration plus 3 times the standard deviation of the duration. 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. A flow in a trace is 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 rate of flows that bring more trouble to routers (big, long, heavy, bursty).

But the correlation among these metrics was not clear. By relying on data recorded from two different operational networks (Los Nettos and NLANR) the authors in [22] studied flows in four different dimensions, namely size, duration, rate and burstiness, and examine how they are correlated. This paper analyzes the relationships between the different heavy-hitters and concludes that there is a big (over 80%) correlation between Cheetahs and Porcupines, between Elephants and Cheetahs and between Elephants and Porcupines, while there is a small correlation between Tortoises and all the others. A few of the exceptions seem to be related to network attacks (multiple TCP SYN retransmissions), which seem to be a good suggestion for a future work identifying network attacks based on the detection of these types of flow. In summary, they made three 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 for the network operator to gather information from the available network traffic records. In [23] the authors present three techniques for the identification of elephant flows. The first two single-feature techniques (aest and α-constant load) result in highly volatile elephant classification over time. The third one 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. They argue that such approach is more successful in isolating elephants that exhibit consistency – high volume over long time.

B. Traffic Modeling

Many research papers that deal with network traffic modeling issues start with a traffic analysis approach before proposing any analytical approach. Therefore, we can take advantage of this procedure to gather some 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) [24] correlations and non-Gaussian marginal distributions. In our context, it is important to know that in a typical aggregate traffic model, traffic bursts arise from several simultaneously

active connections.

In [25], the authors developed a new framework for analyzing and modeling network traffic that moves 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 this 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 give rise to burstiness and heavy-tailed connection durations give rise 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 be the transient response to re-routing, the transient response to start/stop of connections, the TCP slow-start peculiarities or the heterogeneity in bottleneck links for passing flows.

In a general analysis of Internet traffic, Kim et al [26] performed a flow-based investigation on four 1Gbps networks 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 on this paper. First, they confirm that the relation between the number of bytes and the duration of a flow is very week, 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, ignoring flash flows will eventually improve the performance and accuracy of flow/volume prediction systems. Finally, the paper concludes that ignoring flash flows will eventually improve the performance and accuracy of flow and traffic prediction systems. This conclusion is somewhat controversial, since flash flows greatly influence the level of congestion in a network and their presence should not be overlooked.

The concept of flow aggregation through Internet links is presented in [22]. It describes a method 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 are dragonflies, a significant number of flows have lifetimes of hours to days, and can carry a high proportion (10-25%) of the total bytes on a given link. They also define tortoises as flows that last longer than 15 minutes (1-2% of the traffic observed). They point out that flows can be classified not only by lifetime (dragonflies and tortoises) but also by size (mice and elephants), and observe that flow size and lifetime are independent dimensions. As usual, the authors argue that Internet Service Providers (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 (LR) flows can contribute a significant fraction of their packet and byte volumes – something they may not have allowed for when using traditional 'flat rate user bandwidth consumption' approaches to provisioning and engineering.

Following the same approach of his previous work in [22], Brownlee [27] analyzes network flow lifetimes for a backbone link in two different sites. It studies the effect of the flow capture strategy that involves discarding the short-lived flows, referred to as dragonflies. Long-lived flows are called tortoises. The author observed that a high proportion of traffic bytes are carried in tortoise flows. Brownlee suggests that by 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, with non-dedicated hardware.

C. User Behavior

Due to the increased complexity and processing power required for performing behavior analysis from packet traces and the restrictions on payload data usage in force in some 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 characteristic of volume statistics. Packet-level analyses are forbidden in some countries, and therefore must be used only when necessary and possible. Flow-level analyses economize processing resources and have 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. 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 examples are understanding seasonal

[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

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

traffic volumes, user connection durations, traffic growing trends, packet arrival processes, self-similar [28] (fractal) behavior and traffic matrix estimation. This information has been used both by Internet Service Providers (ISPs) for network dimensioning and resource provisioning and by the Internet research community for an in-depth understanding of the Internet traffic 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].

To conclude this section we describe two recent research studies that evaluate user behavior from a cable modem-based service provider. It is worth stressing that while there are many papers that address Internet measurements, there are few research papers with an in-depth traffic analysis of broadband users.

In [29], 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 Line (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 consist 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 was a first step to construct a generalized parametric model for broadband access networks.

In [30], Marques et al first separate users' traffic in two main categories, namely residential and business. They analyzed user session arrivals and durations and concluded they 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 also pointed out that business users tend to stay connected longer than residential ones, while residential users tend to connect to more often. 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, namely 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.

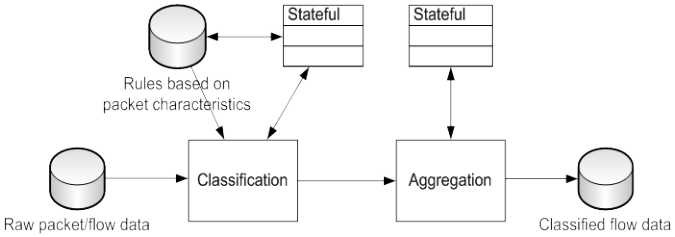

Fig. 3: Packet Classification Methodology.

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. use data structures that describe an application behavior (e.g., graphlets, 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.

A. Inference Quality Metrics

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

Completeness is a metric over the coverage of a detection technique. Detection completeness is measured in percentage (of the expected detection sample count), and may be over 100% (when more flows are detected than what should be). A low detection completeness indicates 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.

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 application, these extra detected flows are false positives and their presence diminishes the value of accuracy.

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 valid, a technique must have completeness near 100% and a high accuracy, also near 100%.

B. Goals on Traffic Classification

Accurate traffic classification is fundamental to numerous network activities [31][32], including:

- Identification of application usage and trends: the correct identification of user application 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: for ISPs, knowing the applications their subscribers are using may be of vital interest for application-based accounting and charging or even for offering new products. For example, 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 port number [33]. This method uses only packet headers (or flow identifiers).

In the most common method 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 [33]. 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 carry more and more traffic that uses dynamically allocated port numbers. Further, many applications can 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 with 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 lot of work to implement every protocol 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. 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 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* (a.k.a. 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 ratio of false positives and many times only experiences with traffic traces provide enough feedback to select the best performing byte signatures. E.g., the above example with the '\GET' signature finds both HTTP and Gnutella, thus it is not proper 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. 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 [34], Sen et al provide an interesting investigation on payload-based P2P traffic classification. They analyzed the most common P2P applications and concluded 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 [35][36][37]. All of these methods need a reference packet level trace and reference traffic classification.

Another question of interest is the efficiency of signature matching. Basic regular expression checks are very processor intensive and slow. There are papers that focus on this problem [38] and use as promising techniques for pattern matching as the application of bloom [39].

In *statistics based classification* some statistical feature of the packet-level-trace is grabbed and used to classify the network traffic. E.g., a 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 be an indication of a P2P application, which generates 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 cannot be determined in general: e.g., it cannot be stated that a flow belongs to Skype or MSN Messenger voice traffic 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 another way is to automatically discover the features of a specific kind of traffic. To achieve this, 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 [32][40][41]. For some papers regarding skype detection attempts, see [42][43].

In order to perform 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 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 different type of traces where data volumes, transport speeds, traffic mix varies the methods may have to be retrained.

Bernaille et al [44][45][46] 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.

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 [47], 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 [47] 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 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 with packet based approaches. 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 use special knowledge or statistical analysis.

In [31], Karagiannis et al present a novel approach to classify traffic flows into according to the application groups, e.g. P2P, based on connection patterns. Connection patterns are described by graphs, where nodes represent IP address and port pairs and edges represent flows between source and destination nodes.

Connection patterns are analyzed at three levels of detail: (i) the social, (ii) the functional and (iii) the application level. This approach operates in the dark, having (a) no access to packet payload, (b) no knowledge of standard port numbers and (c) 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.

In [20], Xu et al. present a general methodology for building comprehensive behavior profiles of Internet backbone traffic in terms of communication patterns of end-hosts and services. Relying on data mining and information-theoretic techniques, the methodology consists of significant cluster extraction, automatic behavior classification and structural modeling for in-depth interpretive analyses. 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 [48].

The definition of what is *unwanted traffic* is still very fuzzy and greatly varies among networks, especially from networks of different countries [49]. It is 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 are [49]: SPAM, Denial-of-Service (DoS) attacks, Distributed DoS attacks (DDoS), host misconfiguration, host scanning and host exploiting.

Focusing anomaly detection (not application identification), Soule et al [50] use Origin-Destination flows to generate a traffic matrix and compares four tests (the last two new proposals) to detect anomalies: 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 [52][51][52] 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 [53] 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 (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 reference. This reference must be considered trustable. This can be achieved by manual classification of traffic traces 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 every day new applications appear. Due to similar reasons, it is essential to validate the classification approaches on measurements from more networks.

Another option would be to generate the traffic that will be analyzed, therefore knowing the exact applications, but this also requires a long time and does not allow a comparison with real traffic: there is no guarantee that results will correspond to those from a real network.

Due to the problems discussed above, validation is a difficult task. Comparing the accuracy and completeness of the algorithms based on different measurements with potentially different reference classification is not straightforward.

A. Comparison of traffic classification methods

The results of the BLINC method [31] are summarized in Table II. The authors develop their own byte signatures and use them for validating the proposed connection pattern based classification method. As shown in Table II, the Blinc algorithm is able to recognize the main application types. The algorithm performs better in terms of accuracy than completeness.

Table II Accuracy and Completeness of the Blinc Method

Application	**Metric**	**Blinc [31]**
WWW	completeness	69-97%
	accuracy	98-100%
Mail	completeness	78-95%
	accuracy	85-99%
Bulk transfer (FTP)	completeness	95%
	accuracy	98%
Chat	completeness	68%
	accuracy	98%
Network Management	completeness	85-97%
	Accuracy	88-100%
P2P	completeness	84-90%
	Accuracy	96-98%

The evaluation of the Bayesian method [32] is listed in Table III. 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. The accuracy of the P2P traffic classification is lower than in case of other applications. This is in line with the fact that P2P applications are diverse and their main characteristics are difficult to grab.

Table III Accuracy of the Bayesian Method

Application	**Bayesian [32]**
WWW	99.27 %
Mail	94.78%
Bulk transfer (FTP)	82.25%
Services	63.68%
Database	86.91%
Multimedia	80.75%

P2P	36.45%

The method proposed by [45] also uses machine learning, but the algorithm reads only the first few packet headers in each connection. The accuracy of the classification methods is summed up in Table IV. The authors use payload analysis as a reference. 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 IV ACCURACY OF THE ON-THE-FLY ALGORITHM

APPLICATION	ON THE FLY [45]
HTTP	99%
HTTPS	81.8%
SMTP	84.4%
POP3	0%
POP3S	89.8%
BULK TRANSFER (FTP)	87%
NNTP	99.6%
KAZAA	95.24%
EDONKEY	84.2%
SSH	96.92%

Basically, the tables above show that the algorithms perform well on the analyzed traces. On the other hand, the fact that all proposed methods use heuristics implies that some fine tuning work may be needed to fit the methods to other traces or new applications.

One way of getting around of 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 [47]. This approach also has the advantage that mismatching classification results are recognized automatically. Furthermore, such traffic may be dumped separately for further analyses and the knowledge gained can be incorporated into the algorithms.

Authors of [47] executed the different classification methods on different measurements one-by-one, 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 [47] Table 1. In this work the comparison of a byte signature based analyses and a method that combines the results of many previously cited methods is shown in Table V.

TABLE V ACCURACY AND COMPLETENESS OF THE SIGNATURE-BASED ANALYSIS

Application		Payload Analysis [47]
Web	Accuracy	91%
	completeness	134%
P2P	accuracy	99%
	completeness	61%
Chat	accuracy	97%
	completeness	76%
E-mail	accuracy	97%
	completeness	78%
File Transfer	accuracy	99%
	completeness	26%
Streaming	accuracy	98%
	completeness	3%
System	accuracy	95%
	completeness	75%
Tunneled	accuracy	10%
	completeness	120%

The joint application – and the comparison run on the same input data – shows that some methods are stronger in accuracy and others provide more complete results (see [47] Table 1 for details). This is illustrated in Fig. 4. Strict protocol parsers are the least complete while heuristic based approaches are the most complete. Regarding the accuracy, the order is reversed, thus signature based approaches are more accurate than simple heuristic based approaches.

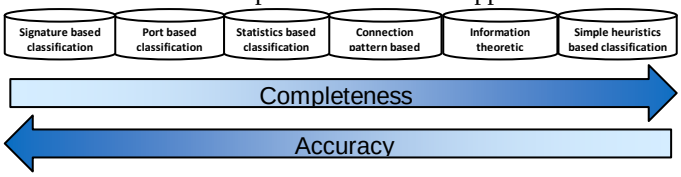

Fig. 4: Measured accuracy and completeness of different techniques

As a consequence, the application classification decision is a trade-off between the amount of traffic left unknown and the bigger chances of erroneous classification.

VI. CONCLUSIONS

Many techniques for network management and application identification do exist, but some have legal problems (signature-based payload analysis) and 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 cannot be utilized in real-time in high-speed links. 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 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, filtering and billing. The next sections present some recommendations for traffic classification and list some open research questions.

A. Requirements for Application Classification Analyses

Despite the importance of traffic classification, an accurate method that can reliably address this problem is still to be developed. The state of the art in the identification of network applications through traffic monitoring relies on the use of well known ports: an analysis of the headers of packets is used to identify traffic associated with a particular port and thus of a particular application. However, 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.

The alternative method of identifying applications, which means inspecting the packet payload, has some known drawbacks, such as hardware and complexity limitations (i.e., lack of scalability), privacy and legal issues and payload encryption. 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

Some questions in traffic measurement and analysis for application identification still remain open, and are discussed here.

- 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.
- 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. Is it possible to create a better approach for identifying applications from NetFlow records?
- All application classification methods leave parts of the traffic unclassified. Therefore a general goal is to reduce the amount of unclassified traffic.
- Today the evaluation of classification methods measures classification error in terms of traffic volume. This implies that as long as a classification algorithm recognizes big size flows, its evaluation will show good results. On the other hand, there are other important traffic characteristics that may not depend on volume, e.g. flow count. Consequently, existing application classification methods need to be tested also in this respect. Further research should focus on classifying small size flows as well.
- Application classification methods use heuristics. As a consequence of that, their validation will always be problematic. 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. One extreme of this is to mark traffic for which the classification is certain.
- Updating heuristics is time consuming. This motivates the research of methods that can recognize new protocols or changes in protocol versions of the same application type automatically. In addition, 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.
- New applications sometimes impose changes in network capacity and management (e.g., Voice over IP usage, Video on Demand, Internet Worms). Network operations can not presently promptly react to traffic changes caused by new applications. Could there be a way around it, for fast new-application profiling? Or should the network manager have the option to penalize new applications (with QoS) until their traffic profile is thoroughly understood?
- Some providers use only a single method for network management. Others use many, including traffic measurements and signature-based analyzers. A few

develop their own, programming new management applications or developing simple scripts. Some network operators manage only border gateways (incoming and outgoing traffic), while others also pay great attention for inside traffic. Can the necessary infrastructure for traffic management, including a traffic measurement and analysis sub-part, be described in general, or is it only possible on a case-by-case study?

- 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 longer hours 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 these users. 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?

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.

APPENDIX

TABLE VI ACRONYMS AND ABBREVIATIONS

Name	Definition
AS	Autonomous System
BFW	Broadband Fixed Wireless
DoS	Denial of Service
DSL	Digital Subscriber Line
IPFIX	IP Flow Information Export
ISP	Internet Service Provider
LRD	long-range-dependent
P2P	Peer-to-Peer
QoS	Quality of Service
SNMP	Simple Network Management Protocol
SOHO	Small Office Home Office
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

REFERENCES

[1]
[2] Azzouna, Nadia Ben; Guillemin, Fabrice; “Analysis of ADSL traffic on an IP backbone link”, IEEE Globecom 2003.
[3] Sullivan, Mark, “Surveys: Internet Traffic Touched by Youtube”, Light Reading, http://www.lightreading.com/document.asp?doc_id=115816, January 2007.
[4] Crovella, Mark; Krishnamurty, Balachander, “Internet Measurement: Infrastructure, Traffic and Applications”, book, ISBN-13 978-0470014615, Wiley, 2006.
[5] Cho, Kenjiro; Fukuda, Kenshue; Esaki, Hiroshi; et Kato, Akira; “The Impact and Implications of the Growth in Residential User-to-User Traffic”, ACM SIGCOMM 2006.
[6] Marques Nt., H. T. et al., “Characterizing Broadband User Behavior” ACM Workshop on Next-generation Residential Broadband Challenge, 2004.
[7] Cherry, S., “The VoIP Backlash”, IEEE Spectrum Magazine, October 2005, http://spectrum.ieee.org/oct05/1846
[8] Orebaugh, A., Morris, G., Warnicke, E. & Ramirez, G., “Ethereal Packet Sniffing”, Syngress Publishing, February 2004.
[9] Cisco IOS NetFlow, “Introduction to Cisco IOS NetFlow - A Technical Overview”, White Paper, Last updated: February 2006, http://www.cisco.com/en/US/products/ps6601/products_white_paper0900aecd80406232.shtml.
[10] Sadasivan, G.; Brownlee, N.; Claise, B.; Quittek, J.; “Architecture for IP Flow Information Export”, IP Flow Information Export WG (expires in March 2007), September 2006.
[11] Duffield, N., & Lund, C., “Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure”. ACM Internet Measurement Conference 2003.
[12] Duffield, N.; Lund, C.; Thorup, M. “Estimating Flow Distributions from Sampled Flow Statistics”, e ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.
[13] Duffield, N.; Lund, C.; Thorup, M., “Learn more, sample less: control of volume and variance in network measurement”, IEEE Transactions in Information Theory, vol. 51, no. 5, pp. 1756-1775, 2005.
[14] Fernandes, S.; Kamienski, C.; Kelner, J.; Sousa, D.; and Sadok, D., A Stratified Traffic Sampling Methodology for Seeing the Big Picture”, Submitted to The International Journal of Computer and Telecommunications Networking, 2007.
[15] Sommer, R. and Feldmann, A. “NetFlow: information loss or win?”, In Proceedings of the 2nd ACM SIGCOMM Workshop on internet Measurement (Marseille, France, November 06 - 08, 2002). IMW '02. ACM Press, New York, NY, 173-174.
[16] Liu, Yong; Towsley, Don; Ye, Tao and Bolot, Jean, “An Information-theoretic Approach to Network Monitoring and Measurement”, Internet Measurement Conference, IMC '05, 2005.
[17] 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.
[18] Paxson, Vern; “Bro: A system for detecting network intruders in real-time” Computer Networks, vol. 31, no. 23-24, pp. 2435-2463, 1999.
[19] 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.
[20] Xu, Kuai; Zhang, Zhi-Li; Bhattacharya, Supratik, “Profiling Internet Backbone Traffic: Behavior Models and Applications”, ACM SIGCOMM 2005.
[21] Lan, K. and Heidemann, J. 2006. “A measurement study of correlations of internet flow characteristics”. Computer Networks 50, 1 (Jan. 2006), 46-62.
[22] Brownlee, N.; Claffy, K.C., "Understanding Internet traffic streams: dragonflies and tortoises", Communications Magazine, IEEE, vol.40, no.10, October 2002, pp. 110-117.
[23] Papagiannaki, K., Taft, N., Bhattacharyya, S., Thiran, P., Salamatian, K., and Diot, C. 2002. “A pragmatic definition of elephants in internet

backbone traffic". In Proceedings of the 2nd ACM SIGCOMM Workshop on internet Measurement (Marseille, France, November 06 - 08, 2002). IMW '02. ACM Press, New York, NY, 175-176.
[24] Park, Kihong & Willinger, Walter, "Self-Similar Network Traffic and Performance Evaluation", Wiley-Interscience, ISBN: 978-0471319740, First Edition, January 2000.
[25] Sarvotham, S., Riedi, R., and Baraniuk, R. "Connection-level analysis and modeling of network traffic". In Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (San Francisco, California, USA, November 01 - 02, 2001).
[26] Kim, Myung-Sup; Won, Young J.; Hong, James W.; "Characteristic analysis of internet traffic from the perspective of flows", Computer Communications Journal, 2005.
[27] Brownlee, Nevil; "Some Observations of Internet Stream Lifetimes", Lecture Notes in Computer Science, Volume 3431, January 2005, pp. 265 – 277.
[28] Willinger, Walter et al, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level", IEEE/ACM Transactions on Networking, volume 5, pp. 71-86. February 1997.
[29] Sinha, A.; Mitchell, K. and Medhi D.; "Flow-Level Upstream Traffic Behavior in Broadband Access Networks: DSL versus Broadband Fixed Wireless", IEEE IPOM, 2003.
[30] Marques, H. T., Rocha, L. C., Guerra, P. H., Almeida, J. M., Meira, W., and Almeida, V. A. "Characterizing broadband user behavior", Proceedings of the 2004 ACM Workshop on Next-Generation Residential Broadband Challenges (October 15 - 15, 2004). NRBC '04. ACM Press, New York, NY, 11-18.
[31] Karagiannis, T., Papagiannaki, K. & Faloutsos, M., "BLINC: Multilevel Traffic Classification in the Dark", ACM SIGCOMM 2005, August/September 2005.
[32] Moore, Andrew & Zuev, Denis, "Internet traffic Classification Using Bayesian Analysis Techniques", Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, ACM Press, 2005, pp. 50-60.
[33] IANA, "Port Numbers", http://www.iana.org/assignments/port-numbers
[34] Sen, Subhabrata; Spatscheck, Oliver; Wang, Dongmei; "Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures", Proceedings of the 13th international conference on World Wide Web, 2004, pp. 512-521.
[35] Haffner, P.; Sen, S.; Spatscheck, O.; Wang, Do., "ACAS: Automated Construction of Application Signatures", MineNet 2005.
[36] Kim, H. A.; Karp, B., "Autograph: Toward Automated, Distributed Worm Signature Detection", Security 2004.
[37] Li, Z.; Sanghi, M.; Chen, Y.; Kao, M.-Y.; Chavez, B., "Hamsa: Fast Signature Generation for Zero-day Polymorphic Worms with Provable Attack Resilience", S&P 2006.
[38] Markatos, E. P.; Antonatos, S.; Polychronakis, M.; Anagnostakis, K. G., "Exclusion-based Signature Matching for Intrusion Detection", CCN 2002.
[39] Erdogan, O.; Cao, P., "Hash-AV: Fast Virus Signature Matching by Cache-Resident Filters", Globecom 2005.
[40] Zander, S.; Nguyen, T.; Armitage, G., "Automated Traffic Classification and Application Identification Using Machine Learning", IEEE LCN 2005.
[41] McGregor, A.; Hall, M.; Lorier, P.; Brunskill, A., "Flow Clustering Using Machine Learning Techniques", PAM 2004.
[42] Bonfiglio, Dario; Mellia, Marco; Meo, Michela; Rossi, Dario; Tofanelli, Paolo, "Revealing Skype Traffic: when randomness plays with you", ACM SIGCOMM 2007 Data Communication Festival (SIGCOMM 2007), Tokyo, Japan, August 2007.
[43] Suh, Kyoungwon; Figueiredo, Daniel R., Kurose, Jim, Towsley, Don, "Characterizing and Detecting Skype-Related Traffic", The 25th Conference on Computer Communications. Barcelona, Spain, April 2006.
[44] Bernaille, Laurent; Teixeira, Renata; Salamantian, Kavé; "Early Application Identification", Second Conference on Future Networking Technologies, December 2006.
[45] Bernaille, Leurent; Teixeira, Renata; Akodkenou, Ismael; Soule, Augustin; Salamantian, Kave; "Traffic Classification On The Fly", ACM SIGCOMM Computer Communication Review, Volume 36, Number 2, April 2006, pp. 23-26.
[46] Bernaille, Laurent; Teixeira, Renata, "Early Recognition of Encrypted Applications", in Proceedings of the 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, April 2007.
[47] Szabo, G.; Szabo, I.; Orincsay, D., "Accurate Traffic Classification", WOWMoM 2007.
[48] Silvestre, G.; Fernandes, S.; Kamienski, C.; Sousa, D.; Sadok, D., "Padrões de Comportamento de Tráfego P2P", Proceedings of the III Workshop of Peer-to-Peer (WP2P 2007), June 2007.
[49] Andersson, L.; Davies, E.; Zahng, L.; "Report from the IAB workshop on Unwanted Traffic", Internet-Draft, IETF Network Working Group, February 2007.
[50] Soule, A.; Salamantian, K.;, Taft, N.; "Combining Filtering and Statistical Methods for Anomaly Detection", Proceedings of the Internet Measurement Conference 2005, pp. 331-344, 2005.
[51] Huang, C.-T.; Thareja, S.; Shin, Y.-J.; "Wavelet-based Real Time Detection of Network Traffic Anomalies", Proceedings of Workshop on Enterprise Network Security (WENS 2006), August 2006.
[52] Magnaghi, A.; Hamada, T.; Katsuyama, T.; "A Wavelet-based Framework for Proactive Detection of Network Misconfigurations", Proceedings of the ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality, pp. 253-258, 2004.
[53] Lakhina, Anukool; Crovella, Mark; Diot, Christophe; "Mining Anomalies Using Traffic Feature Distributions", ACM SIGCOMM 2005.
[54] Chhabra, Parminder; John, Ajita; Saran, Huzur, "PISA: Automatic Extraction of Traffic Signatures", 4th International IFIP-TC6 Networking Conference, May 2005.
[55] Iannaccone, G., "Fast Prototyping of Network Data Mining Applications", 7th Passive and Active Measurement Workshop. Adelaide, Australia, March 2006.
[56] Jiang, Hongbo; Moore, Andrew; Ge, Zihui; Jin, Shudong; Wang, Jia, "Lightweight Application Classification for Network Management", SIGCOMM Workshop on Internet Network Management (INM), August, 2007.
[57] Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.C., "Transport Layer Identification of P2P Traffic", ACM SIGCOMM 2004, October 2004.
[58] Karagiannis, T.; Papagiannaki, K.; Taft, N.; 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.
[59] Kundu, Sumantra; Pal, Sourav; Basu, Kalyan; Das, Sajal, "Fast Classification and Estimation of Traffic Flows", in Proceedings of the 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, April 2007.
[60] Tai, Masaki; Ata, Shingo; Oka, Ikuo, "Fast, Accurate, and Lightweight Real-time Traffic Identification Method based on Flow Statistics", in Proceedings of the 8th International Conference, Passive and Active Measurement Conference, Louvain-la-Neuve, Belgium, pp. 255 -299, April 2007.

Arthur de C. Callado received the degree of Bachelor in Computer Science in 2000 from the Federal University of Ceará in Fortaleza, Brazil and the degree of Master in Computer Science in 2004 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 and also pursues a PhD degree in the same institution.

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

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 is a professor at the Federal Center for Education in Technology of Alagoas and also a senior researcher in the Network and Telecommunications Research Group from the Federal University of Pernambuco.

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.

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.

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.