Intelligent Network Client Profiler

Diogo Teixeira and Artur Arsénio

Abstract—Peer2Peer traffic already accounts for a large share of the overall internet traffic. Future solutions will need to manage all the available resources in order to charge users using fair rules according to their communication profile. Obtaining information about the behavior of Internet traffic is therefore fundamental to the management, monitoring and operation activities, such as the identification of applications and protocols that customers use. However, the main obstacle to this identification is the lack of scalability for monitoring network devices. In particular, they can analyze all the network packets for this purpose. This task is extremely demanding and almost impossible to accomplish rapidly in large networks (because usually there is a number in the hundreds or thousands of customers). Furthermore, we expect such networks to become even larger, as on the internet of things all devices (sensors, appliances, etc) will be publicly connected to the internet. As such, traffic sampling strategies have been proposed to overcome this major problem of scale. This paper presents different works in the area of monitoring traffic for user profiling and security purposes. It proposes as well a solution that uses selective filtering techniques combined with an engine traffic DPI to identify applications and protocols that customers use most frequently. Thus it becomes possible to get ISPs to optimize their network in a scalable and intelligent manner, imposing security measures in order to enforce network usage according to client profiles.

Index Terms—Adaptive sampling, client profiling, deep packet inspection, intelligent networks, selective filtering.

I. INTRODUCTION

Today, IP networks are used in virtually all areas, from the Internet, companies, private and personal level, linking all types of devices we can imagine: computers, printers, smart phones, game consoles, televisions, sensors, etc. The proliferation and growth of Internet Protocol (IP) networks (e.g. Internet), coupled to the explosive growth and booming worldwide of the number of devices connected to them, led IP networks to be used in various and multiple purposes. Internet Service Providers (ISPs), in parallel with this growth, have faced many problems at various levels, particularly related to security concerns, as well as the unfair usage of IP networks by a small set of users.

Due to the large increase of users and critical applications, and the emergence of new network attack techniques, computer networks have brought new challenges for their managers. The networks need stability because they are responsible for transporting massive amounts of information

and data from multiple sources, many of them confidential. Thus, it is common to attempt to attack networks, aiming to disrupt the confidentiality, integrity and/or availability of data and information.

Associated with the aforementioned growth, there is also an increase in network complexity. In this context, not only becomes increasingly difficult to understand the dynamics of these heterogeneous networks at large scales, as the configuration and maintenance reaches very high and demanding levels of complexity [1].

A. Traffic Congestion

A major and undesirable factor associated with this proliferation, which impairs the performance of the network, is traffic congestion, which is a major concern for packet-switched networks such as IP networks. Traffic congestion occurs whenever the amount of traffic entering a portion of the network per unit of time, is larger than that network elements' capability to process and/or forward such traffic, giving rise to full buffers at the routers and to consequent packet loss. The network congestion puts into question the availability of bandwidth on the network, a problem that over time has worsened, despite numerous mechanisms, technologies and algorithms developed to ensure quality of service. The overload and network congestion is due to the profusion of services and applications with implications for traditional usage models, such as various peer-to-peer (P2P) applications and technologies [2], and the increasingly common usage of multimedia sites for video sharing (e.g. YouTube), life casting websites and live video streaming (e.g. Ustream) as well as the proliferation of online games (e.g. World of Warcraft), among others. Voice over IP (VoIP) is also nowadays a predominant service on the internet, posing restricted requirements concerning traffic delay and jitter. And there is increasingly a stronger consumer demand for converged network services.

With the large increase in the future on the number of IPv6 devices, such as sensors, connected to the internet, it is expected further large increases in traffic leading to network congestion. The appearance of new interactive and personalized multimedia content will require less broadcast and more unicast connections from terminals to content providers, further increasing network traffic very significantly. Furthermore, usage of P2P technologies for multimedia transmission will prejudice the network control by the telecommunications operator. Hence, network congestion will be a significant problem. Congestion is often due to failure to adjust the traffic flow to the excess in the packet arrival rate relative to service rate in one or more network nodes. This excess leads to the imbalance in traffic at the network nodes, where a set of resources is overloaded, while another set of resources may be underutilized [3].

Manuscript received December 19, 2012; revised March 7, 2013.

D. Teixeira is with Nokia Siemens Networks and Instituto Superior Técnico (e-mail: diogo.teixeira.ext@nsn.com).

A. Arsénio is with the Instituto Superior Técnico (e-mail: artur.arsenio@ist.utl.pt). His work was supported in part by a Harvard-Portugal Collaborative Research Award HMSP-CT/SAU-ICT/0064/2009.

The dominant thought in the past was that congestion could be solved only with a significant increase of transmission speeds in the channels, with an increase in processing power of communication nodes and the use of large buffers for storing packets. However, the authors in [4] show that these procedures alone are not an efficient solution. Currently, the occurrence of congestion on the Internet is mainly due to the unpredictable and chaotic nature of traffic flowing through it. Basically, the current congestion control mechanisms can be classified into two groups: the first increases the availability of resources through the dynamic reconfiguration of the same; and the second, the most used, reduces the demand regarding the availability of resources [5].

B. Automatic and Intelligent Network Security

The Internet is naturally a constantly changing environment, which enables its customers to create and adapt their technology, according to their needs and desires. This mutation becomes troublesome in certain cases, as criminal activities become associated as well with the emergence of the Internet. The online crime, and consequently the problem of computer security and networking, is another major concern of the current Internet Service Provider (ISP). Illegal and illicit acts are performed over the Internet (e.g. distribution, exchange and sharing of copyrighted material, streaming freely “pay TV channels”, pedophile content exchange, illegal casinos and illegal gambling).

Denial of Service (DoS) attacks are rising, producing disruptive results to operators, and costly network repair and maintenance. Indeed, corporations report currently very high numbers of unauthorized network access. Indeed, most firewalls require every year critical patches.

Given these major concerns on the part of ISPs, it becomes essential to get to know what each user does, that is, the technologies and applications they use. It is also important to determine if the user employs the network to commit criminal acts. It thus becomes clear, that must be created algorithms that analyzes the network traffic generated by users and create user profiles so that ISPs can implement security and business policies according to the profiles generated. Networks have become more complex, and thus the traffic characterization and monitoring are increasingly important tools for traffic engineering, enabling network operators to have diverse information concerning network usage.

Based on the need for a more dynamic and efficient architecture for communication networks, and on the importance of anticipating future problems, there is a need to develop intelligent mechanisms that produce outputs to assist effectively the early decisions by the ISP. These systems must be capable of monitoring in a scalable way a large IP network, and prevent a drop in performance that could degrade the quality of service. It is desired as well to enable the determination that a network client is using the network for illicit acts and with criminal objectives.

It is therefore necessary to “predict” efficiently the variation of traffic and its implications on the next generation IPv6 network’s quality of service and security.

II. OVERVIEW ON INTRUSION DETECTION / PREVENTION

An Intrusion Detection System (IDS) is a security

management tool that aids and automates the process of monitoring network events. It monitors hosts in order to detect suspicious activity through anomalies or signatures and generate alerts without interfering with network traffic. The IDS with active response is aimed at taking action indirectly to automatically shut down suspicious activity detected. IDS alone is not able to stop attacks, it needs the help of other mechanisms to carry out this function. On the other hand, an Intrusion Prevention System (IPS) has the same mechanisms for detection as IDS, but can stop in real time and automatically a suspicious activity with or without the help of other devices.

A. Detection Methods

Both IDS and IPS use three detection methods to generate alerts or block any suspected traffic. The detection methods may be based on anomalies, or on signatures, or hybrid.

Signature-based Detection technology is based on the use of a database for the storage of patterns for certain attacks, which are used to make comparisons with possible attacks that may be occurring. According to [6] a network IDS signature is a pattern that is being searched on the network traffic. When a signature for an attack corresponds to the observed traffic, an alert is generated, or else an event is recorded. The signature of an attack is built based on the characteristics of the packet that contains the attack. Some of them are: source / destination port, sequence numbers, protocol flags (e.g. syn, ack), and especially a small fragment of the application layer.

The disadvantages of this system are, on the one hand, its invasive nature (information privacy issue) and on the other hand, it only detects known signatures and possible variations to them. Thus, it is very important to always keep an updated database of signatures.

Anomaly-based detection assumes that each user has a profile of resource utilization, aiming to detect deviations from these patterns to identify possible attacks. This method consists of identifying different behaviours (anomalies) on a host or a local network. It is assumed that the attacks are different from ordinary activities (legitimate) and that can be detected by the system that identifies these differences. Anomaly detectors build profiles representing normal behaviour of users, hosts or network connections.

The hybrid intrusion detection combines signatures-based and anomalies-based detection methods. It aims to correct the deficiencies presented by each of these two methods.

B. Architectures

The main architectures for IDS or IPS are as follows.

Host-based architecture: the system is installed on a host and it is solely responsible for the host’s safety. This architecture includes HIDS (Host Intrusion Detection System), used to detect and analyse suspicious activity in a particular host [6]; and HIPS (Host Intrusion Prevention System), functioning similarly to HIDS but preventing attacks before they succeed. The big difference for an active response system is that prevention systems have direct access to applications and the Operating System (OS) kernel itself [7]. *Network-based architecture:* one or more IDS or IPS sensors will be responsible for network segments. This architecture

includes NIDS (Network Intrusion Detection System), which aims to monitor all traffic passing through a given network segment to identify threats, by detecting scans, probes and attacks [6]. The NIDS architecture has one or more sensors that are responsible for analyzing the traffic that passes through a given network segment. NIPS (Network Intrusion Prevention System) is an inline device, i.e., it is located directly in the packets path as they traverse the network [7]. Usually, host based IDS/IPS are mainly targeted to internal users, whereas network based IDS/IPS are mainly targeted for outside intruders, although there are exceptions.

Distributed architecture: this architecture can have local and network sensors. This architecture includes DIDS (Distributed Intrusion Detection System) and DIPS (Distributed Intrusion Prevention System);

DIDS and DIPS can have local sensors (HIDS) and / or network sensors (NIDS). This architecture differs mainly from others on the fact that in addition to having a management station to which the sensors are connected, the sensors can also exchange information among themselves. The rules for each sensor can be customized depending according to needs [8].

C. IPsec with ESP

The usage of IPsec with ciphering, namely through the employment of ESP (Encapsulating Security Payloads) on IPsec protocol suite, can make IDS and IPS job more difficult. Indeed, ESP provides services such as confidentiality, integrity, and data origin authentication, providing the capability to scramble TCP packets. This may block the IDS/IPS from deciphering the TCP packets for monitoring. Depending on the application in mind, this can pose severe limitations to IDS/IPS usage.

III. NETWORK TRAFFIC MONITORING

Monitoring network traffic is an essential activity for the active or passive management of networks. It can be performed through observation of packets or flows. Significant research efforts aim to understand deeply how the traffic characteristics of various applications affect the network infrastructure behaviour. Thus, measurement strategies are essential to identify abnormal behaviours (e.g., sudden high volume of generated traffic). Network monitoring can be based on SNMP (Simple Network Management Protocol), and also based on NetFlow and sFlow protocols, designed specifically for traffic analysis.

A. Packets Sampling and Flows Sampling

Despite the monitoring activity has become usual with the help of existing tools in routers (e.g. Cisco NetFlow), several problems still persist. The main current obstacle for measurement (packets or streams) based traffic monitoring is the lack of scalability relative to links capacity. In other words, monitoring traffic on links with very large capacities leads to creation of huge volumes of data [9]. As the capacity of links and the number of flows increases, maintaining counters for each flow crossing routers becomes expensive both at the computational and economic levels [10].

Therefore, several sampling strategies have recently been

proposed as a way to optimize the selection of packages (for accounting flows) [9], [11], [12] or flows selection (for statistical analysis of the original traffic) [13]. The simple sampling process (uniform) does not provide adequate results because the IP flows generally follow Pareto distributions, also known as long-tailed distributions (the long tail), for their packet and bytes [14]. Some existing sampling techniques are dependent on flow size in which only are accounted relatively larger flows.

It is therefore essential to rigorously explore and analyze the various existing approaches with respect to the sampling of packages.

B. Stratified Sampling

In this section, we describe the stratified sampling technique applied to the traffic analysis and its use for reducing the volume of sampled data. In stratified sampling [15], a population of N units is first divided into sub-populations of sub-units. Sub-populations (called strata) do not overlap and together cover the entire population. After determining the strata, a sample shall be selected in each of them, being the selections made separately in different strata. When simple accidental samples are selected in each stratum, the whole process is called accidental stratified sampling. Stratification is a common technique that can provide increased precision in estimates of the characteristics of the entire population [15]. In general, it is possible to divide a heterogeneous population into isolated sub-populations that are homogeneous. If all strata are homogeneous in the sense that the value of the measures vary slightly from one unit to another, one can obtain an accurate estimate of the average value of any one stratum, given a small sample of that stratum [15].

Finally, these estimates can be combined to form an accurate estimate of the total population. Stratified sampling can be classified as uniform, proportional or Bowley, and optimal. In uniform stratified sampling all strata have the same size, while on proportional sampling the number of elements in each stratum is proportional to the size of the stratum. Finally, optimal stratified sampling considers, besides the size of the stratum, the variability within the stratum [15].

According to [16], stratified sampling is an example of a hybrid technique. The basic idea behind the stratified sampling is to increase the accuracy of the estimate, using a priori information about the correlations of the characteristics investigated with some other trait easier to obtain. The a priori information is used to perform an intelligent grouping of the elements of the main population. So, a better estimate can be obtained with the sample size, or it can be even possible to reduce the sample size without reducing the estimate accuracy. Many articles address this sampling mode. In [17] the authors explore this method as a tool for describing the traffic behaviour traffic at the flow level. In [18, 19] it is used cluster analysis techniques (i.e. K-Means Clustering and Large Applications - CLARA), and its application to stratified sampling of traffic flows. The results presented in [18] and [19] clearly show that the algorithms CLARA and K-Means are suitable for the realization of stratification based on the "duration of flows" metric. Article [19] shows how the

estimation accuracy can be improved without increasing the size of the sample using stratified sampling techniques. Throughout the article are investigated different stratification strategies with regard to the possible reduction of the number of packages sampled. It is shown that the sample size can be significantly reduced if packets are stratified according to their dimensions.

C. Adaptive Sampling

The sampling rate translates (directly or indirectly) the accuracy of the estimation process. Some network behaviours (e.g. anomalies) may not be detected accurately using low sampling rates. On the other hand, high sampling rates produce large amounts of data that must be submitted and processed later for the collector. In periods of high traffic, network equipment can not cope with the required sampling rate and discard excess packets. The increase in the number of samples may influence the total traffic, since mostly sampled packets are sent via UDP. Thus, it is important to avoid congestion. It is therefore obvious that there is a trade-off between accuracy and performance. Choosing the best sampling rate is also a challenging task.

Network traffic exhibits variability in the number of packets that traverse the links in different periods. What is distinctive about the network behaviour are sudden bursts of traffic, since the network traffic can be characterized by a heavy-tailed distribution [20]. Briefly, in situations of inactivity (or low load), the samples should be more widely spaced in order to reduce the bandwidth consumed and the amount of information stored. Whenever there is a significant network activity, sampling should be more frequent so as not to lose important information about the status and performance of the network (although this represents an additional consumption of bandwidth).

A technique that can be used to perform adaptive sampling is linear prediction [21], which uses past samples to estimate future measurements. This technique is grouped with a set of rules that define the adjustments to be carried out on the sampling interval, according to the feedback on correct or incorrect predictions. According to [20] the linear prediction provides sufficient accuracy when compared with other methods. Another alternative technique is Fuzzy logic, which uses past sampling to calculate the various parameters of the algorithm, and thereby automatically adjust the sampling interval.

In terms of mechanisms for adaptive sampling, a vast group of solutions is focused on implementing customized sampling methods. The article [22] describes two adaptive sampling methods to manage the processor usage on a network device. One method uses information about the current usage of the processor to adjust the sampling rate. The other method uses the packets arrival times (which can be used to anticipate a traffic burst) together with knowledge about the processing time required to process a sample. [23] proposes to use the least squares estimate and a certain set of heuristic rules to determine the sampling rate. The authors in [24] describe a flow sampling approach, which allows controlling the expected volume of samples and minimizes the variance of the estimates. The proposed smart sampling method adapts the sampling process, combining the likelihood of a flow

being selected with the flow size. This process shifts the focus to the "elephant" flow, which has a severe impact in the volume of traffic. The authors in [16] include on their solution a sampling rate adaptability mechanism. [25] proposes an adaptive sampling method, which adjusts the sampling rate so as to limit the error of the estimate of the flow volume without oversampling. The method allows controlling the estimate accuracy, being a tradeoff between the measurement utility and overhead.

The authors in [25] introduce sticky sampling, a means to adapt the sampling rate based on the number of records stored. Mechanisms to dynamically change the sampling rate for each flow, in order to maintain uniform relative error, are provide in [26]. Many other approaches exist on the prediction and adaptation of the sampling rate, which differ substantially both in accuracy and in complexity. One of the simplest solutions is the naive forecast, which assumes that the number of packets to the next interval would be equal to the number of packages for the current period. It is a solution that requires virtually no computation, but the accuracy of the estimates is missing. Most forecasting methods already discussed may be used in this case. However, access to some data from the device internal network is not possible (as the state of the queues, packet arrival rate, resource utilization in real time, etc.)

D. Deep Packet Inspection

Deep Packet Inspection (DPI) involves a thorough analysis of packets that traverse the network, examining not only the header, as done by Shallow Packet Inspection (SPI), but also their content. However, the packets from the Internet are not only formed by the payload data added by a single header. Indeed, at each layer of the multi-tier architecture, there is a header added to the load, and the payload header contains a layer of the upper layer. Therefore, a better definition is based on the border between the IP header and IP payload.

Thus, the definition of Deep Packet Inspection can be given as the act of any network equipment (excluding terminal equipment on the endpoint of a communication channel) using any field on a layer on top of the network layer, in contrast to the SPI, which only checks the portion of the header of a packet [26]. Modern network devices employ deep packet inspection for the implementation of sophisticated services, such as intrusion detection and prevention, traffic shaping, load balancing, firewalls, spam detection, virus, among others. The deep packet inspection is a powerful mechanism to perform matching criteria on packages [26].

A technical report [20] has indicated that Snort can be considered as a software DPI, but it is not able to handle high-speed traffic. This is mainly due to the limitations of sequential Van Neumann architecture and also to poor optimization of the regular expressions used for the match. In the case of deep packet inspection it is often necessary to match the patterns to each byte offset. Most likely many signatures must be compared with the packet payload. Thus, the process requires a large number of comparison operations. The sequential treatment is not suitable for this mode of operation and for this reason customized parallel approaches are being employed. A DPI should be able to provide at least

the number of standard index and information about the location. It should also support the grouping of patterns. A group of patterns should only be tested if the packet belongs to that group. A more efficient and accurate DPI system can be constructed using a header classifier running in conjunction with a payload matcher.

As the sequential architecture is not adequate to perform the DPI task, some researchers focused on developing parallel implementations of FPGA (Field Programmable Gate Array). Usually one of the following three algorithms is used for efficient multi-patterns matching: (a) Bloom Filter algorithm, (b) Aho-Corasick, (c) Boyer-Moore algorithm; being (a) and its extensions the most common algorithms. This algorithm uses multiple hash functions and can produce false positives (but never a false negative). In the article [26], it is implemented and tested six different ways of sampling packets (i.e. invariable random sampling, invariable mechanical sampling, random time sampling, mechanical time sampling, random sampling speed mode and speed mechanical sampling) in a DPI to detect streams of P2P data in high availability networks.

IV. ARCHITECTURE FOR AUTOMATIC CLIENT PROFILING

We have presented several tools and protocols important to gather samples of network traffic, process such information, in order to be able to make decisions concerning security, and services usage in general.

This section provides a scalable architecture that is capable of inferring the behavior of a particular network client (user) through the analysis of sampled network traffic, by association with a usage profile. It is proposed an intelligent and scalable algorithm for real-time traffic analysis. The system automatically configures the network based on policies, in order to meet the following requirements:

- Real-time traffic analysis and capture;
- Optimal sampling rate Estimation for package capture;
- Segmenting (organizing) traffic captured by customer;
- Mapping a client to a particular profile.

Fig. 1. Network Security architecture for User Profiling. The solution is composed by two main components: the analyzer, to select the sampling rate and sampling modes, and to effectively sample the traffic; and the Centralized Server, responsible for inspecting further the sampled traffic, implementing prediction and client profile mapping.

A user profile is composed by a set of services/technologies

that users use most often (i.e. P2P sites, video sharing, online gaming sites, etc). The use of such services produces traffic, which is somehow behind the degradation of network performance, contributing to its congestion. Through each profile, there will be one or more policies associated (i.e. application of reduced bandwidth, traffic shaping, policy-based consumption, etc.) Fig. 1 represents the proposed architecture, which components, and its functions, are overviewed hereafter.

A. Analyzer

The analyzer is the entity responsible for i) traffic sampling; ii) inspecting each sampled packet; iii) insertion into the database of the data packet inspection output.

This component should be implemented in a second aggregation router (on the ISP network). The second aggregation routers are the equipment installed after the Digital Subscriber Line Access Multiplexer - DSLAMs (Fig. 2). In general, second-generation routers, usually called EDGE Routers, have large storage capacity and processing. The analyzer consists of the entities: Sniffer; Sampling Module; DPI Module; and Database, described hereafter.

The "sniffer" is responsible for intercepting all traffic passing through the router, and passing it (copying it) to the sampling module. The "sampling module" is responsible for sampling the traffic delivered by the sniffer, and it consists of three components: i) Sampling Modes; ii) Sampling mode selection; iii) Sampling rate selection.

There is the need to choose multiple sampling modes, because there is no single mode that will, by itself, match all diverse and unpredictable traffic variations. With three sampling modes: systematic sampling time-based, Random sampling, and Stratified random sampling, we can tailor the best sampling, depending on various conditions.

Fig. 2. Analyzer location. This component will be implemented in a second aggregation router (on the ISP network). In general, second-generation routers, usually called EDGE Routers, have large storage capacity and processing.

The component "sampling mode selection" asks the mode to be loaded to the component immediately above. The mode that will be charged is calculated using three variables: traffic volume, router computational load, and connection speed. To find out which mode to be loaded should be used a learning algorithm (e.g. a neural network), which will be trained

according to the three input variables. The component "sampling rate selection" is responsible for finding the appropriate pace to sample the traffic, being also responsible for sampling the traffic when it is found the correct sampling rate (using a prediction technique). After being sampled, the traffic goes to the next module: module DPI, which is responsible for inspecting each sampled packet, and for identifying what technology / protocol it matches. The DPI engine holds the result of each packet inspection in a database.

The database can be formed according to the format shown in Fig. 3. The values in the figure represent the number of packets that have been identified for client X, of technology Y (e.g. BitTorrent, eDonkey, SOPCast, World of Warcraft).

	Technology 1	Technology 2	Technology 3	.	.	.	Technology N
Client 1	350	357	7	.	.	.	5566
Client 2	456	0	456	.	.	.	0
Client 3	324545	357	2343	.	.	.	62
Client 4	5545	2457	0	.	.	.	27
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
Client N	2345667	123246	666	.	.	.	69

Fig. 3. Database #1 Format example.

B. Centralized Server

A "central server" is responsible for processing the data obtained by the Analyzer, as well as implementing prediction. The central server consists of the following entities, in addition to databases:

The interviewer, which has the following functions:

- Interviewing after T time (e.g. 24 hours), all the databases of various analyzers implemented in the first two aggregation routers, and replicate the data to the database # 2.
- Acts as SNMP Manager, interrogating all routers (including an SNMP agent) requesting the number of packets that passed by them at the last time period t (e.g. 1 minute), and saves this information in the database # 3.

The data processor, which has the following functions:

- Prepare the data to be analyzed by the next entity (i.e. Mapping Profiles).
- Responsible for implementing the prediction technique.

The prediction technique is relatively simple to implement. The "interviewer" functions as SNMP agent and asks periodically all routers on the number of packets that passed through it during a specific time interval. The interviewer puts all the information in the database # 3. The "data processor" uses this information to inform component "sampling rate selection" on the expected traffic for that time interval (e.g. one hour). The "data processor" recalculates at the end of a longer time period (e.g. each week and each month) the statistics of the traffic volume that occur in each router, thus being able to predict with greater accuracy, the volume of traffic on the route.

The entity "mapping profiles" is a neural network. Its input data are the number of packages of technologies that the customer used more significantly, or the percentage of total traffic of each client's technology, for the neural network to learn the correct customer profile to associate a client.

V. CONCLUSIONS

This paper proposed a scalable architecture that is able to learn the usage pattern that each client makes of the network, and for what purpose (or purposes) the customer uses it. Customers are associated with a particular usage profile. Each user profile is associated with pre-defined policies that will allow an automatic optimization of the network. The architecture presented uses more effective sampling methods. Performing selectively the sampling of packages becomes the most efficient way to circumvent the scalability problem, so that one does not have to analyze all network packets to perceive and understand the characteristics of network traffic. From a sample of traffic, one can get and learn the characteristics of the original traffic so that the sampling mode is effective. And such data is then used to learn client network usage profiles.

REFERENCES

- [1] L. Ho, C. Macey, and R. Hiller, "A distributed and reliable platform for adaptive anomaly detection in IP networks," in *Proc. the 10th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management: Active Technologies for Network and Service Management*, 1999.
- [2] R. Schollmeier, "A definition of Peer-to-Peer networking for the classification of Peer-to-Peer architectures and applications," in *Proc. the First International Conference on Peer-to-Peer Computing, IEEE*, 2002.
- [3] D. Awduche, "MPLS and Traffic Engineering in IP Networks," *IEEE Communications Magazine*, 1999.
- [4] R. Jain, "Congestion control in computer networks: Issues and trends," *IEEE Network Magazine*, 1990.
- [5] V. Jacobson and M. Karels, "Congestion avoidance and control," *ACM SIGCOMM, Symposium on Communication Architectures and Protocols*, 1988.
- [6] S. Northcutt, L. Zeltser, S. Winters, K. Fredrick, and R. Ritchey, "Inside network perimeter security: The definitive guide to firewalls, VPNs, routers, and intrusion detection systems," *New Riders Publishing*, 2003.
- [7] A. Orebaugh, M. Rash, J. Babbin, and B. Pinkard, *Intrusion prevention and active response: Deploying network and host IPS*, Syngress Publishing, 2005.
- [8] J. Beale, J. C. Foster, J. Posluns, R. Russell, and B. Caswell, "Snort 2.0 intrusion detection," *Syngress*, 2003.
- [9] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *Proc. the ACM SIGCOMM 2003*, 2003.
- [10] C. Eitan and G. Varghese, "New directions in traffic measurement and accounting," in *Proc. the ACM SIGCOMM 2002*, 2002.
- [11] N. Hohn and D. Veitch, "Inverting sampled traffic," in *Proc. ACM Internet Measurement Conference - IMC'03*, 2003.
- [12] N. Duffield, "Sampling for passive internet measurement: A review," *Statistical Science*, 2004.
- [13] S. Fernandes, T. Correia, C. Kamienski, D. Sadok, and A. Karmouch, "Estimating properties of flow statistics using bootstrap," *IEEE MASCOTS 2004*, 2004.
- [14] N. Duffield, C. Lund, and M. Thorup, "Charging from sampled network usage," in *Proc. the ACM SIGCOMM Internet Measurement Workshop*, 2001.
- [15] C. Kamienski, T. Souza, S. Fernandes, G. Silvestre, and D. Sadok, "Caracterizando propriedades essenciais do tráfego de redes através de técnicas de amostragem estratificada," 2005.
- [16] T. Zseby, M. Molina, F. Raspall, N. Duffield, and S. Niccolini, "Sampling and filtering techniques for IP packet selection," *IETF RFC5475*, 2009.
- [17] N. Duffield, D. Chiou, B. Claise, A. Greenberg, M. Grossglauser, and J. Rexford, "A framework for packet selection and reporting," *IETF RFC5474*, 2009.
- [18] S. Fernandes, C. Kamienski, D. Mariz, and D. Sadok, "Avaliação de Técnicas de Agrupamento na Amostragem de Tráfego na Internet," *24th Brazilian Symposium on Computer Networks (SBRC 2006)*, 2006.

- [19] S. Fernandes, C. Kamienski, D. Mariz D. Sadok, and J. Kelner, "A stratified traffic sampling methodology for seeing the big picture," *The International Journal of Computer and Telecommunications Networking*, vol. 52, issue 14. Elsevier North-Holland, Inc. New York, NY, USA 2008.
- [20] R. Jurga and M. Hulbój, "Packet sampling for network monitoring," Technical Report, 2007.
- [21] T. Zseby, "Stratification strategies for sampling-based non-intrusive measurements of one-way delay," *Passive and Active Measurement Workshop Proceedings*, 2003.
- [22] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," *IETF RFC 2330*, 1998.
- [23] S. Chaudhuri, R. Motwani, and V. Narasayya, "On random sampling over joins," in *Proc. the 1999 ACM SIGMOD International Conference on Management of Data*, 1999.
- [24] N. Duffield, C. Lund, and M. Thorup, "Learn more, sample less: Control of volume and variance in network measurement," *IEEE Transactions in Information Theory*, 2005.
- [25] G. Manku and R. Motwani, "Approximate frequency counts over data streams," in *Proc. the 28th International Conference on Very Large Databases*, 2002.
- [26] H. Chen, F. You, X. Zhou, and C. Wang, "The study of DPI identification technology based on sampling," *Information Engineering and Computer Science*, 2009.

Diogo Teixeira was born in Cascais, Portugal on July 9, 1986. He concluded the Degree (BSc) in Communication Networks Engineering in 2009. Now is an undergraduate Master Degree (MSc) in Communication Networks Engineering in Instituto Superior Técnico (IST), Lisbon, Portugal. His primary

research interests are network management, traffic engineering, traffic and packet Sampling and automatic network configuration. He is also R&D Engineer at Nokia Siemens Networks (NSN) on TNMS (Telecommunication Network Management System), a well-established, feature-rich product for element layer and network layer management of optical transport networks.

Artur M. Arsenio received his doctoral degree in Computer Science from Massachusetts Institute of Technology (MIT) in 2004, and he was a researcher at Rodney Brooks' Humanoid Robotics group at MIT CSAIL. He received both his MSc (1998) and Engineering Degrees (1995) in Electrical and Computer Engineering from Instituto Superior Técnico (IST), in Lisbon. He is the inventor of 6 international patent families, and authored/co-authored more than 90 scientific publications. He is the recipient of several scientific and innovation awards.

He is Y dreams Robotics CEO, an innovative company aiming at turning connected things into robots. He is also Assistant Professor in Computer Science at IST. Before, he headed Innovation at Nokia Siemens Networks Portugal SA. He also worked as Senior Solution Architect leading several international R&D teams for Siemens, and as Chief Engineer represented Siemens Networks on IPTV Standardization forums. He collaborated on the creation of the interdisciplinary Institute for Human Studies and Intelligent Sciences in Cascais, being the AI Lab co-director. He is a co-founder and associate member at Beta-i entrepreneurship association, having mentored dozens of start-up projects, and he was a co-organizer of TEDxEdges, the first TEDx event ever in Portugal. He is co-founder and vice-chair of ACM SIGCOMM Portugal chapter, a Fulbrighter, and President of the MIT alumni association in Portugal.