Detection and Classification of Peer-to-Peer Traffic: A Survey

JOÃO V. GOMES, PEDRO R. M. INÁCIO, MANUELA PEREIRA, and MÁRIO M. FREIRE,
University of Beira Interior and Instituto de Telecomunicações
PAULO P. MONTEIRO, Nokia Siemens Networks, University of Aveiro and Instituto de Telecomunicações

The emergence of new Internet paradigms has changed the common properties of network data, increasing the bandwidth consumption and balancing traffic in both directions. These facts raise important challenges, making it necessary to devise effective solutions for managing network traffic. Since traditional methods are rather ineffective and easily bypassed, particular attention has been paid to the development of new approaches for traffic classification. This article surveys the studies on peer-to-peer traffic detection and classification, making an extended review of the literature. Furthermore, it provides a comprehensive analysis of the concepts and strategies for network monitoring.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*; *Packet-switching networks*; C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management; Network monitoring*; C.4 [**Computer Communication Networks**]: Performance of Systems—*Measurement techniques*

General Terms: Management, Measurement, Security

Additional Key Words and Phrases: Application classification, deep packet inspection, behavioral analysis, peer-to-peer, traffic monitoring

ACM Reference Format:
Gomes, J. V., Inácio, P. R. M., Pereira, M., Freire, M. M., and Monteiro, P. P. 2013. Detection and classification of peer-to-peer traffic: A survey. ACM Comput. Surv. 45, 3, Article 30 (June 2013), 40 pages.
DOI: http://dx.doi.org/10.1145/2480741.2480747

1. INTRODUCTION

In the early years of the Internet, network connections relied on the *client-server* paradigm, generating an asymmetric amount of data in both upstream and downstream directions. Nonetheless, users became more influent, not only on the information available on Internet, but also on its distribution. The so-called Web 2.0 offered Internet hosts the opportunity to provide their own multimedia contents and to directly interact with other peers. Furthermore, the popularity gained by peer-to-peer (P2P) systems in the end of the last century enabled the direct distribution and sharing of contents

This work was partially supported by the *Instituto de Telecomunicações*, by University of Beira Interior, and by *Fundação para a Ciência e a Tecnologia*, through the grant contract SFRH/BD/60654/2009 and the project TRAMANET: Traffic and Trust Management in Peer-to-Peer Networks with contracts PTDC/EIA/73072/2006 and FCOMP-01-0124-FEDER-007253.
Authors' addresses: J. Gomes, P. Inácio, M. Pereira, and M. Freire, Instituto de *Telecomunicações*, Department of Computer Science, University of Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal; emails {jgomes, inacio, mpereira, mario}@penhas.di.ubi.pt; P. Monteiro, Nokia Siemens Networks Portugal, S. A., Rua Irmãos Siemens, 2720-093 Amadora, Portugal; email: paulo.1.monteiro@nsn.com.

DOI: http://dx.doi.org/10.1145/2480741.2480747

between Internet users. The once passive user has gained a new and very active role in the Internet, acting simultaneously as client and server. These important changes in the services running over the Internet and in the behavior of the end-hosts modified the traditional properties of network traffic, which is evolving towards a more balanced bandwidth usage in both directions. Additionally, most of these applications present a greedy profile, consuming as much bandwidth as they can, which may end up interfering with priority policies. Azzouna and Guillemin [2003], for example, found that 49% of the traffic in an asymmetric digital subscriber line (ADSL) link was caused by P2P applications, while Gerber et al. [2003] and Sen and Wang [2004] observed the growth and prevalence of this kind of traffic. In 2007, *ipoque* conducted a world wide study about Internet traffic [Schulze and Mochalski 2007], and the results showed that P2P file-sharing applications were producing more traffic than all the other applications together, being responsible for 49% to 83%, on average, of all Internet traffic and reaching peaks of over 95%. Another study by *ipoque* [Schulze and Mochalski 2009], in 2008 and 2009, concluded that although the total amount of traffic generated by P2P file-sharing has increased, its percentage has decreased to an average value of between 42.51% and 69.95%. This fact may be due to an increase of traffic generated by video streaming and file-hosting Web services, like YouTube, Tudou, or RapidShare. Yet, there have been several discussions regarding the adoption of P2P solutions by some of the these services, namely YouTube and Tudou, in order to accelerate their downloading rates and reduce the transmission cost. In fact, the Web-based CNN live channel service relies now on the P2P paradigm due to a plugin each user has to install.

In spite of the share of global traffic of each Internet application, P2P systems motivate particular attention from the perspective of network management for the dual role their peers play. For a certain amount of data downloaded by a peer, a portion of data is also uploaded by the same peer. Instead of being concentrated in a dedicated server, the distribution cost of the service is thus shared by the users. While this fact is advantageous for content providers, it implies that a host receiving a service will produce additional traffic in its Internet service provider (ISP) network or local area network (LAN), as it is also providing the service to a different peer. Moreover, hosts in P2P networks usually receive and provide contents from and to several peers at the same time. Hence, P2P applications are likely to produce a much larger number of connections than typical client-server applications. In addition, mechanisms for searching contents in remote peers also cause an increment of the communications between hosts. These facts make P2P traffic management more challenging than traffic from client-server applications, which is usually formed by a single or a few connections. Besides, of the increase of the bandwidth consumption, the amount of traffic generated by P2P applications in both directions is more balanced, as opposed to the greater weight in downstream of the traditional client-server traffic. This difference poses an important issue in terms of traffic management, as most networks (or Internet connections) were devised to offer lower bandwidth in upstream. Managing the network and implementing specific policies for P2P traffic does not necessarily means it should be blocked or heavily throttled. Nevertheless, there are techniques that can help to efficiently manage this traffic if one is able to classify it, as content caching [Karagiannis et al. 2005b; Xu et al. 2008].

Although the traffic management issues are of particular concern mainly for ISPs and network administrators [Karagiannis et al. 2005b; Freire et al. 2009], there are other problems, mostly related to security risks and vulnerabilities [Zhou et al. 2005; Seedorf 2006; Li et al. 2007; Johnson et al. 2008, 2009; Chopra et al. 2009] that are magnified by the distributed nature of P2P systems and by the role of their peers, and that may affect companies and home users. While reducing the overlay distances between end hosts

for the exchange of contents, the P2P paradigm also amplifies the effects of viruses and other threats by facilitating their dissemination. Ensuring privacy, anonymity, or confidentiality is also more difficult in these networks and constitutes a real concern, not only for home users but also for companies [Lawton 2004]. These problems do not result directly from the P2P communication paradigm, but they are a consequence of the proximity between peers and of the simplicity of content sharing in P2P systems. This fact, together with the multiple connections created by P2P applications and the encryption and obfuscation techniques used by most of them, makes it more difficult to identify threats in the traffic.

In this context, traffic classification based on the application protocol appears as a crucial tool for managing the data within the networks, fairly sharing the available bandwidth, assuring the quality of service (QoS), implementing billing mechanisms, or deploying security measures. However, identifying the application that generated the traffic is nowadays a difficult task and may have several associated issues (e.g., random port number or payload encryption), as described by Kind et al. [2008]. The traditional and most obvious method for classifying network traffic was to associate the transport port numbers to well-known application protocols. However, this approach became ineffective as soon as a significant number of applications started to use random port numbers or port numbers used by other well-known protocols. Karagiannis et al. [2004a] identified P2P applications running on port 80 and estimated that 30% to 70% of the overall P2P traffic is generated by applications using random port numbers. Likewise, the results by Madhukar and Williamson [2006] show that the same percentage of Internet traffic cannot be correctly identified by port-based methods. More recently, Basher et al. [2008] concluded that 90% of the P2P traffic may be using random ports.

Therefore, in the last years, the classification of Internet protocol (IP) traffic has been a very active research field, with many contributions based on distinct approaches. When port-based mechanisms lost their effectiveness, the solution was to employ deep packet inspection (DPI) techniques, which were frequently used by network intrusion detection systems (NIDSs) for security purposes, to identify the traffic using signatures in the contents of the packets. However, this approach also has a few important drawbacks, mainly related with computational resources requiring to inspect traffic in high-speed networks, with the impossibility of accomplishing their purpose when the payload is encrypted and with privacy issues. The alternative was to design different statistical or behavioral (based on heuristics) methods, which resort to the packet header and flow-level data to segregate the traffic into different classes.

The main contribution of this article is to survey the existing studies, methods, techniques, and applications on the topic of traffic classification. Although several concepts and techniques may also apply to other fields of traffic monitoring, herein they will be analyzed from the perspective of traffic classification. Most of the classification methods may be applied to the classification of the traffic from different types of applications. Nonetheless, since P2P systems are on the basis of a large number of research contributions, a special attention will be given to the studies addressing the subject of P2P traffic classification.

In order to facilitate the understanding of the survey, an introduction it is included to the subject of network measurement from the perspective of traffic monitoring (and, more specifically, classification), which explains a few important concepts and techniques. The existing approaches for traffic classification are also carefully described in the survey, explaining their way of functioning, in which situations they are more valuable, and what their limitations are. Foremost, this article provides an extended review of the literature, presenting the available methods and their performance and organizing them based on the type of analysis they perform.

The remainder of the article is structured as follows. Section 2 describes the related work. Section 3 gives an explanation of important concepts and techniques for traffic measuring, while Section 4 describes the distinct approaches for traffic classification together with their main advantages and weaknesses. An analysis of the published literature is presented in Section 5, followed by conclusions.

2. RELATED WORK

The topic of traffic classification has aroused considerable interest in recent scientific contributions, with several studies addressing the challenges raised by new application protocols and proposing novel techniques and solutions for its classification. Nonetheless, there are still few papers surveying the existing works on the field as well as analyzing distinct methods and approaches.

The Internet Measurement Research Group (IMRG) of the Internet Research Task Force (IRTF) sponsored a workshop on *Application Classification and Identification*. The report of this workshop [Strayer et al. 2008] described a number of important topics, highlighting the challenges inherent to the task of traffic classification and its main motivations and summarized the contribution of each paper.

Madhukar and Williamson [2006] compared the efficiency of three distinct techniques for the identification of P2P traffic: port numbers, payload signatures, and transport-layer heuristics. In order to provide a longitudinal study of the performance of each technique, they collected traffic traces during two years and used them as sample data to evaluate each method. Kim et al. [2007, 2008] also performed a comparative study, between three different approaches to traffic classification: port-based, behavioral, and statistical. The evaluation was based on available applications and research tools and techniques: *CoralReef* [Moore et al. 2001], *BLINC* [Karagiannis et al. 2005a], and machine learning (ML). The authors tested the solutions using seven distinct traffic traces from two backbone and two edge links from the U.S. Japan, and Korea. Li et al. [2009] compared four different classification methods in terms of efficiency and effectiveness: well-known port numbers, DPI, Naïve Bayes, and the *C4.5* decision tree method. In order to evaluate the performance of the mechanisms from both temporal and spatial perspectives, the authors used traffic traces collected over several years on two different sites.

A survey on traffic classification solutions relying on ML was provided in Nguyen and Armitage [2008b]. Although the study was especially focused on the identification of application-level protocols through the use of ML techniques, the authors also included a description of the difficulties imposed by many recent Internet applications and the main reasons for developing new methods for the classification of the traffic generated by those applications. Cascarano et al. [2010b] compared the performance of three different traffic classifiers for peer-to-peer television (P2PTV) applications: a DPI mechanism, a method based on single-class support vector machines (SVMs), and a method based on multiclass SVMs. They evaluated three P2PTV applications and used traffic traces collected at the border gateway of a LAN of a university campus.

The closest work to the study presented herein was the one by Callado et al. [2009]. After introducing the subject of traffic analysis, the authors described the state of the art of flow-based traffic analysis, pointing out several flow properties of Internet traffic. They also described many research works on the traffic classification field and provided a theoretical comparison of the results obtained by four distinct studies.

This survey distinguishes itself from the previous works for its wide and comprehensive analysis and for giving special attention to the identification of P2P traffic and its challenges. Moreover, as traffic classification is a very active research topic, many works described herein are subsequent to Callado et al. [2009]. Unlike most studies, this survey starts by introducing the subject of traffic measurement from the

perspective of traffic classification, so that basic concepts (important for the correct reading of the remainder of the article) can be well understood. Besides describing the existing approaches for traffic classification and identifying its main advantages and weak points, this survey provides a broad review of the literature. Furthermore, it analyzes, compares, and gives a structured view of studies, approaches, techniques, and available applications for the classification of P2P network data.

3. MEASURING FOR NETWORK MONITORING

Solid research studies on the characteristics and behavior of computer networks, as well as the development of effective mechanisms for the traffic management and the design of better and more efficient networks, require strong and accurate traffic analyses and collections. Over the last few decades, many authors have addressed the subject of network (and, more specifically, Internet) traffic measurements, highlighting its crucial role for understanding the behavior of computer networks (e.g., [Jain and Routhier 1986; Claffy and McCreary 1999; Cáceres et al. 2000; Williamson 2001; McGregor 2002; Paxson 2004]).

However, measuring network traffic is far from a simple problem. Corroborating this idea, Paxson [2004] describes a few challenges one has to deal with when performing such task, as well as some interesting strategies for a sound Internet measurement. McGregor [2002] also discusses several technical issues, while proposing guidelines for quality measurements.

Likewise, in the context of traffic classification, and in spite of playing an essential role in a solid work, network measurements can be a source of technical challenges [Arlitt and Williamson 2007]. In the next sections, the topic of traffic measurement is explored from the point of view of traffic classification, considering important concepts, techniques, and approaches. Nevertheless, for a deeper discussion on the subject of network measurement, we refer to the book by Crovella and Krishnamurthy [2006], as well as to the references cited in this section.

3.1. Traffic Measurements

At this point, it is useful to distinguish between different approaches for network traffic measurement or monitoring. Based on a few specific characteristics, Williamson [2001] classifies the research tools for network study into the following categories: *hardware* or *software*; *protocol level*; *LAN* or *Wide Area Network (WAN)*; *online* (or real-time) or *offline*; and *passive* or *active*. The discussion of each of these categories may be appropriate or not, depending on the purpose of each monitoring study or tool. However, in most studies [Claffy and McCreary 1999; Paxson 2004; Duffield 2004; Bartlett et al. 2007b], authors differentiate, mainly, between active and passive measurements. Herein, these aspects will be briefly discussed from the perspective of traffic classification.

3.1.1. Hardware- and Software-Based Solutions. Practitioners and researchers working in the field of traffic classification are more interested in analyzing the IP packets or the Ethernet frames. Hence, it is not significant if the traffic measurements are made using hardware- or software-based tools.

Nonetheless, dedicated hardware solutions tend to present a better processing performance, which is useful for real-time analyses. A few companies, like *ipoque* [2011], *Endace* [2011], *Napatech* [2011], or *WildPackets* [2011], provide hardware systems for traffic monitoring or high-speed network interfaces with dedicated buffers for traffic capturing, like the data acquisition and generation (DAG) cards. In terms of traffic classification, a few authors also resort to hardware devices, like field-programmable gate arrays (FPGAs) or ternary content addressable memory (TCAM), to improve the computational performance of DPI mechanisms [Yu 2006; Mu et al. 2007].

3.1.2. Protocol Level. It is possible to measure the traffic at different, and even multiple, protocol levels. However, since traffic classification is mostly used for Internet traffic, measurements for that purpose are usually made at the Ethernet or IP levels.

3.1.3. LAN and WAN. For the purpose of traffic classification, measurements can be conducted, with no loss of information or research knowledge, in LANs instead of WANs, which typically are not so easy to access.

3.1.4. Online and Offline Analyses. Although in terms of the traffic measurement, online and offline approaches do not differ significantly, the latter is more used whenever a real-time analysis is not necessary, since such a task would require higher computational power to be accomplished in high-speed links. Moreover, the usage of offline trace files is crucial for research and validation purposes, as it allows one to run different analyses through the same data and compare the obtained results.

Nevertheless, online measurements are obviously imperative for, for example, NIDSs, firewalls, or other devices responsible for traffic management, which need to take immediate actions (e.g., drop or forward packets) on the network traffic. However, in these cases, the use of online measurements may impact the performance of high-speed networks.

3.1.5. Active and Passive Measurements. The active approach resorts to the injection of actual packets into the network in order to observe the behavior of the network, hosts, or applications. This kind of measurement is mainly used for monitoring the performance of the network or to identify weak points in the system, being especially suitable for the evaluation of QoS levels. `ping` and `traceroute` are simple examples of tools that implement active measurements.

Since active methods rely on the use of artificial traffic, they allow one to easily control the simulation of the scenarios that he or she wants to analyze or to test, like the traffic class, nature, frequency, etc. However, such traffic will not directly reflect the behavior or the influence of the application and of the human behavior. Moreover, these methods will increase the traffic load in the network, which may affect not only the available bandwidth, but also the performance of routers, switches, or other network equipment. In the case of large networks, administrators can face scalability problems when using active measurement techniques.

Passive measurement techniques do not produce any additional traffic. Instead of injecting packets into the network, a passive monitor simply looks at the traffic and collects data that can be used to infer on the behavior of hosts, applications, network performance or even on the user influence in the generated traffic. It does not send additional data to the network being monitored, modify any contents, interfere in the packets route (unless it has also other functions, as firewall, gateway, etc.), or increase the traffic load. Furthermore, an important advantage of this kind of approach is that the final data reflects the properties of the real traffic. Passive measurements are, therefore, particularly useful for traffic management, as they retriev important knowledge about the behavior of the traffic.

Nevertheless, passive measurements may produce large amounts of data, which may require ambitious computational resources not only to store and handle that data, but also to process it and generate useful conclusions. For the same reason, its analysis in real time may be a demanding task. In some contexts and for some purposes, the usage of real traffic may also raise a few legal issues [Ohm et al. 2007].

3.2. Per-Packet and Per-Flow Analysis

Measurements made for the purpose of Internet traffic analysis are mainly focused on IP packets or Ethernet frames. The traffic under analysis is usually captured and stored

on a packet-by-packet manner, as the most obvious method to accomplishing the task of capturing traffic is to simply catch each individually data unit traveling in the network. Some of the existent tools for network management include means to display, process, statistically analyze, or even make decisions on each packet individually. This *per-packet* approach is especially interesting for applications like NIDSs (e.g., *Snort* [2010] or *Bro* [2010]), which need to process and decide upon each packet. Also, sniffers or protocol analyzers especially designed for offline analysis, like *Wireshark* [2010] or *Ettercap* [2010], usually inspect each packet deeply, gathering information from all the layers of the protocol stack.

Although packets are individual data units when traveling through the network, a relation exists between many of them [Jain and Routhier 1986]. Usually, they are generated by the same request or application, contain acknowledgement messages from reliability mechanisms (like with Transmission Control Protocol (TCP) traffic), or are simply carrying an amount of data that is too large to fit in a single Ethernet frame. Therefore, the relation between the packets comprises a relatively hidden knowledge about the network and the traffic behavior, which can be assessed by analyzing the traffic in terms of data flows.

A flow is, most of the time, defined as a set of packets that share a common key: source and destination IP addresses and transport port numbers [Claffy and McCreary 1999; Duffield 2004; Duffield et al. 2005; IETF 2008]. It is considered *active* while the time interval between each packet belonging to the flow is lower than a certain threshold. The timeout value may depend on the purpose of the analysis. Although a few studies propose distinct timeouts, Claffy et al. [1995] explored different values and identified 64 seconds as a good compromise between the size of the flow and the effort to initialize and terminate the flows. Furthermore, a flow may also be defined as unidirectional or bidirectional, depending on whether one wants to consider the packets traveling between two address-port pairs in each direction as two independent flows, or the packets in both directions as a single flow [Apisdorf et al. 1996; Claffy et al. 1995]. Because of the usual asymmetry of the traffic exchanged between two addresses in typical client-server connections and also due to the asymmetric routes in the core Internet, unidirectional flows are mostly used in studies on network performance and bandwidth management, for which it is useful to measure the differences in the traffic in both directions [Claffy et al. 1995]. On the other hand, bidirectional flows are a natural option to represent TCP sessions, and for the purpose of traffic classification, they are a more logical approach to follow, as the traffic exchanged between two address-port pairs, in both directions, belongs to the same traffic class and was generated by the same application. Nonetheless, Smith et al. [2001] were able to successfully use unidirectional packet headers traces to analyze TCP transactions.

In order to analyze the traffic from a flow perspective, a monitoring tool can still capture the packets individually, but it has to organize them in a table of flows, based on the source and destination information (address and port). Several tools (e.g., CoralReef [Moore et al. 2001]) were developed to perform flow-based analyses of traffic from network adapters or from offline packet traces. However, it is possible to receive the flow information directly from routers or other network elements (e.g., using a flow export protocol, like *Cisco NetFlow* [2010], or the Internet Protocol Flow Information eXport (IPFIX) [IETF 2008], a standard for exporting flow data currently under development). *NetFlow* data can be read and analyzed by a few existent applications, like *Flow-tools* [Romig et al. 2000] or *FlowScan* [Plonka 2000].

3.3. Collecting Traffic Data

The access to the network data for traffic measuring, as mentioned in a few studies [Duffield 2004; McGregor 2002], may be performed by copying the transmission

signal (e.g., through the use of a splitter) and analyzing it on a dedicated network monitor, by using a router or a switch to copy all the traffic to an output interface, or by directly tapping a shared link. Nevertheless, there are also a few global infrastructures for the active measuring of Internet that collect data from worldwide links [Murray and Claffy 2001]. The datasets containing traffic from computer networks should be carefully handled in order to protect the privacy of the users, as well as other sensitive data. Several considerations and good practices regarding this subject are discussed in Allman and Paxson [2007].

As seen in previous sections, the passive data collection can be made by polling routers to obtain flows data or by packet capturing. While in the former approach, data is usually acquired through the use of protocols like IPFIX, in the latter, the trace files are collected using commercial or public domain network traffic capturing software, like *tcpdump* [2011] and its Windows version, *WinDump* [2011], or even other available tools developed with basis on the *libpcap* [tcpdump 2011] or *WinPcap* [2011] libraries.

Although the most natural means is to capture the complete packet, such technique generates large trace files, which would require larger storage capacity and processing power to handle the traffic in high-speed links. Moreover, the increasing integration of measurement techniques into routers, switches, and other network elements that do not possess a high processing power [Duffield 2004; Jurga and Hulbój 2007] motivates the development of solutions that can reduce the amount of data collected, as described in the next section.

3.4. Trace Reduction

The most common approaches for trace reduction resort to packet filtering or to the minimization of the data that is kept for future analysis [Duffield 2004; Arlitt and Williamson 2007]. It is possible, depending on the specific goals of each study, to monitor exclusively the packets from a given application. However, such selection is usually made using the transport layer port numbers, which is consensually considered a naive approach. Alternatively, one may select only the packets that establish or finalize a connection or a request, or use any other selection criterion that may be more coherent with the objective of a particular analysis and decrease the number of packets to be captured.

The amount of data stored can be reduced by saving a summary of each application protocol-specific request; by capturing a limited portion of the packet or even only the headers of the first layers of the TCP/IP protocol stack; or by keeping information of a flow instead of storing each packet that belongs to it.

A particular case of packet filtering is the use of packet sampling methods [Amer and Cassel 1989], whose objective is to randomly (or pseudorandomly) choose a small set of the packets observed in the measuring point. It is intended that the set of packets obtained be as representative as possible of the traffic one plans to measure. There are different packet sampling techniques which may be more useful in distinct cases, depending on factors like the goal of the study, the network state, the traffic characteristics, or the resources constrains. Jurga and Hulbój [2007] elaborated on the existent methods for packet sampling and their application in network measurements. Duffield [2004] addressed the subject of Internet traffic sampling as well, providing a long and sound structured discussion of several important topics on passive traffic measurement.

4. TRAFFIC ANALYSIS AND CLASSIFICATION APPROACHES

In the early times of the Internet, traffic classification was a straightforward task that was easily accomplished by matching the port numbers of the transport protocols

Table I. Side-by-Side Comparison of the Approaches for Traffic Classification

Approaches	Characteristics	Advantages	Weaknesses
Port number matching	—Associates port numbers with applications	—Low computational requirements —Easy to implement	—Lack of classification performance due to random port numbers
Deep packet inspection	—Relies on payload data	—High classification performance	—May not work for encrypted traffic —Requires high processing resources —Can only be used for known applications
Classification *in the dark*	—Uses only packet header and flow-level information	—Usually lighter than DPI —Applicable for encrypted traffic —Can identify unknown applications from target classes	—Usually has lower classification performance when compared to DPI
Active crawlers	—Based on modified instances of the target applications	—Identifies accurately users of the target applications	—Identifies only the traffic exchanged with the crawler —Injects additional traffic in the network

with the application protocols. However, since many Internet applications—especially the ones based on P2P architecture—evolved to use random port numbers or ports assigned to well-known protocols (e.g., Hypertext Transfer Protocol (HTTP)), identification strategies agnostic to the port numbers became more common. The most natural approach is to look inside the packets and see what type of data they carry and which application protocol was used. Regardless of that, several statistical or behavior-based methods that do not inspect the contents of the packets have been developed more recently. Table I provides a simple side-by-side overview of the main characteristics of each classification approach. For a better understanding of the rest of the article, a discussion on the different types of techniques for traffic classification, the way they operate, their advantages and their drawbacks is provided in the following sections. Furthermore, two additional sections are included to address the topic of ground truth verification and to describe the most common metrics for the evaluation of the performance of a classification mechanism.

4.1. Traffic Classification Based on Port Numbers

The classification of network traffic based on the User Datagram Protocol (UDP) or TCP port numbers is a simple approach built upon the assumption that each application protocol always uses the same specific transport-layer port. This method was mostly useful in the identification of well-known protocols like, for example, HTTP or Simple Mail Transfer Protocol (SMTP), which use the port numbers 80 and 25, respectively. However, many Internet applications easily bypass this identification strategy by simply using random or unknown port numbers, thereby disguising their traffic using port numbers generally used by other well known protocols (e.g., port 80) that are usually allowed by firewalls. Thereby, port numbers as a classification mechanism are considered obsolete [Karagiannis et al. 2004b; Moore and Papagiannaki 2005; Madhukar and Williamson 2006].

```
alert udp $HOME_NET any -> $EXTERNAL_NET any (msg:"LocalRule:P2P eDonkey UDP
outbound - Status Request"; flow:to_server; content:"|E3 96|"; depth:2;
classtype:policy-violation; sid:1000019; rev:1;)
```

Fig. 1. Example of a *SNORT* rule for detecting a payload signature for the traffic generated by *eDonkey* with obfuscation, proposed in Freire et al. [2009].

4.2. Traffic Classification Based on deep packet inspection

DPI methods, usually the most accurate, are based on inspection of the packets' payload. They rely on a database of previously known signatures that are associated to application protocols, and search each packet for strings that match any of the signatures. This approach is used not only in the classification of network traffic, but also in the identification of threats, malicious data, and other anomalies. Because of their effectiveness, classification systems based on DPI are especially significant for accounting solutions, charging mechanisms, or other purposes for which the accuracy is crucial. Figure 1 shows an example of a *SNORT* rule for the detection of a data signature in the traffic from *eDonkey* with obfuscation mechanisms enabled.

However, deeply inspecting each packet can be a demanding task in terms of computation power and may be unfeasible in high-speed networks. Therefore, some mechanisms search only a part of each packet or only a few packets of each flow, as a compromise between efficiency and accuracy. Besides of the performance issues, the inspection of contents of the packet may also raise legal issues related with privacy protection [Ohm et al. 2007].

Nevertheless, the main drawback of DPI techniques is their inability to be used when the traffic is encrypted. Since, in these cases, the contents of the packets are inaccessible (encrypted), DPI-based mechanisms are restricted to specific packets of the connection (e.g., when the session is established) or to the cases when UDP and TCP connections are used concurrently and only the TCP sessions are encrypted. Packets with no payload, which may be malicious, cannot be classified as well. DPI methods are also sensitive to modifications in the protocol or to evolution of the application version: any changes in the signatures known by the classifier will most certainly prevent it from identifying the application. Moreover, DPI methods that rely on signatures for specific applications can only identify traffic generated by those applications.

4.3. Traffic Classification in the Dark

The inspection of the contents of IP packets, as discussed in the previous section, is not always a valid option for the identification of application-level protocols. Therefore, new methods that do not resort to the deep inspection of the packets have been developed. The strategy of this kind of approach, sometimes called *in the dark* [Karagiannis et al. 2005a; Turkett et al. 2008], is to classify the traffic using behavioral or statistical patterns based on flow-level data or generic properties of the packets [Moore et al. 2005], like addresses, ports, packet size, etc.

The main advantage of classification *in the dark* is the ability to identify a protocol without the need to examine the contents of the packet. As a consequence, mechanisms based on this approach cannot aspire to the same accuracy level of DPI methods. Their results should be understood as a strong suspicion regarding the probable application protocol. Nevertheless, recent studies have achieved high success rates in the classification of Internet traffic. Additionally, classification *in the dark* can more easily be applied to unknown applications since many methods based on this approach classify the traffic in classes of applications (e.g., Web traffic, email, video streaming, P2P, etc.) instead of specific applications.

The existent mechanisms use distinct techniques to correlate the traffic properties and conclude on the application protocol, such as statistical measures, sets of

heuristics, or machine learning algorithms. The following sections introduce each of these approaches.

4.3.1. Statistical Mechanisms. Statistical methods usually rely on flow- and packet-level properties of traffic, like flow duration and size, interarrival times, IP addresses, TCP and UDP port numbers, TCP flags, packet size, etc. These properties are used, individually or combined, to calculate statistical values, from simple measures like average or variance, to more complex ones like the probability density function. In some studies [Crotti et al. 2006], statistical models of the traffic from a certain application are created. Generally, such an approach requiress a learning phase in order to build a reference model that can be used to classify unknown traffic.

4.3.2. Heuristics-Based Methods. Many behavioral mechanisms for traffic classification are based on a predefined set of heuristics. Although a large part of them are common to the majority of the research works, distinct combinations or sets are proposed in several studies. Typical heuristics include the network diameter, the presence of nodes acting both as client and server, the number of hosts a user communicates with, the source-destination IP pairs that use both TCP and UDP, the number of distinct addresses and ports a user is connected to, etc. Generally, the set of heuristics is checked sequentially, and depending on the result, the packet (or flow) is classified as belonging (or not) to a certain application-level protocol.

4.3.3. Machine Learning Techniques. A large part of the studies propose classification mechanisms based on different supervised or unsupervised ML techniques, such as Bayesian estimators or networks [Moore and Zuev 2005; Auld et al. 2007], clustering [McGregor et al. 2004], decision trees [Branch et al. 2009], etc. They assemble a set of traffic characteristics which they correlate using probabilistic functions, associating each packet or flow to a certain class.

4.4. Traffic Classification Using Active Crawlers

The majority of the solutions in the literature are passive, as they do not interfere with the data within the network nor do they generate any additional traffic. Nevertheless, some authors have also developed active mechanisms that crawl the network to collect data used to classify traffic [Saroiu et al. 2003]. A few of them implemented fake or modified instances of the target applications whose main purpose is to identify hosts running the original applications [Ohzahata et al. 2005].

This kind of approach is generally used for very specific purposes, such as the identification of users running a certain application. Some authors resorted to active crawlers to collect statistics on the number of hosts running the target application and on the properties of the connections to peers (available bandwidth, latency, etc.) [Saroiu et al. 2003].

4.5. Ground Truth Verification

The use of precollected traffic from computer networks is of critical importance for the creation and testing of new methods for traffic classification with respect to the application-level protocols. Nonetheless, without the ability to assess its ground truth application information, the use of traffic data is of limited value [Sperotto et al. 2009].

The majority of the packet traces publicly available are limited to the headers due to privacy concerns, making it difficult to obtain the associated ground truth regarding the application. For that reason, in most studies on traffic classification, the researchers collect their own traffic traces to test the accuracy of their solutions. Such an approach makes the comparison of different methodologies inconsistent, as the performance of each of them was evaluated in different conditions [Salgarelli et al. 2007]. The

use of methods to accurately verify and label the ground truth information of packet traces before making the headers publicly available would solve the problem while still keeping the private data.

In many studies, the ground truth verification is obtained by using an alternative method as the reference baseline (e.g., port number matching or DPI [Karagiannis et al. 2005a]). However, such an approach will depend on the accuracy of the classifiers used as baseline. Port number matching, for example, is now considered an ineffective option, while DPI may be unsuitable for encrypted traffic. In fact, when a novel mechanism for traffic classification is proposed, under the pretext that the existent solutions are not completely effective, it is nonsensical to test the accuracy of the new method by using an existing one as the baseline for performance comparison.

Alternatively, hand-classification may be used to verify the ground truth information of the traces [Moore and Zuev 2005]. However, the process can be slow and tiresome. Moreover, it is also possible to create traffic collections from a small network of computers, running a predefined set of applications in a controlled environment. Nonetheless, the obtained traces may not exhibit properties that reflect human behavior.

Given the increasing concern regarding this topic, a few authors have more recently addressed the subject of ground truth verification of application traffic. Canini et al. [2009] presented *GTVS*, a framework for improving and simplifying the process of associating traffic data with application-level protocols. It makes use of the packet payload inspection and of multiple heuristic rules to infer the ground truth information, and it provides a graphical interface to facilitate manual verification of traffic traces. Gringoli et al. [2009] proposed *GT*, a toolset for assessing the ground truth of application traffic. Its architecture differs from *GTVS* mainly in the fact that it includes the existence of a daemon, which is supposed to run in each client and return the information of the process that originated each network connection. Although this approach may significantly increase the accuracy of the verification, the deployment of the client daemon may be difficult in most contexts, or even near impossible in large networks. A similar approach was followed by Szabó et al. [2008], who also described a client-based solution. In this case, the authors suggested the implementation of a client driver that inserts a byte mark in each outgoing packet whose size is not yet the size of the maximum transmission unit (MTU), so that it can avoid the IP packet fragmentation.

All these approaches have their merits and weaknesses, but none is perfect. Relying on an alternative classifier to work as the baseline reference enables the ground truth identification in every trace, independently of the size of the network. However, if the reference classifier uses DPI, the payload data in the traces must not be encrypted or removed. Moreover, the evaluation of the performance of the new classification method will always depend on the accuracy of the reference classifier, which may also loose effectiveness when applications evolve and change the properties of their communications (at payload level or even behavioral level). If the results show a certain misclassification rate (even if it is very low), it is impossible to be sure if the error was induced by the new method or by the reference classifier. Of course, this also depends on how challenging the target application is to be classified and on the composition of the traces that are being analyzed. The *GTVS* solution proposed by Canini et al. [2009] is also a strong tool that can significantly improve the task of ground truth identification. Nonetheless, it is based on a combination of different methods for identifying traffic, including DPI, and thus may have similar limitations. On the other hand, although the manual verification of the traffic could allow for better accuracy, it is only feasible for small datasets. The same happens with the approaches that save information during traffic capturing about the process that generated the packets, such as the ones from Gringoli et al. [2009] and Szabó et al. [2008]. This information is very valuable for traffic classification and can help to achieve high accuracy on ground truth. Unfortunately,

the deployment of the client daemons or drivers in all the computers of a large network is also difficult to accomplish. The use of testbeds with smaller networks, in which it is possible to control the applications being used, also allows high accuracy on ground truth identification. However, it may not be representative of the traffic in large-scale networks. The method used to assess the ground truth is extremely important to the quality of the evaluation results. Therefore, one should be aware of the capabilities and limitations of each method when evaluating a classifier.

4.6. Performance Evaluation Metrics

The evaluation of classification methods is made by comparing the results of the classification with the ground truth information of the traces. Each individual case is considered a true positive (TP), true negative (TN), false positive (FP), or false negative (FN) case depending on whether it was correctly classified as belonging to, correctly classified as not belonging to, incorrectly classified as belonging to, or incorrectly classified as not belonging to a certain class.

The analysis of TPs, TNs, FPs, and FNs can be made in terms of packets, flows, or bytes. The evaluation of classification methods based on packets usually presents lower performance, as many packets are similar independent of the application that generates them. For example, a TCP *SYN* packet, used to initiate a connection, is similar for any application. Moreover, many classifiers, especially the ones based on classification *in the dark*, are not design to classify individual packets. The evaluation in terms of flows and bytes may also present different performance levels. In many traces, depending also on the type of traffic they contain, a small number of flows may carry almost all the bytes. The rest of the flows contain only a few small packets. In these cases, if a method correctly classifies only the larger flows, the result of the performance will be very positive in terms of bytes and very negative in terms of flows. On the contrary, if the larger flows are misclassified and the all the small flows are correctly classified, the performance will be positive in terms of flows and negative in terms of bytes.

The performance of the classifiers can be measured, in terms of TPs, TNs, FPs, and FNs, using different metrics [Makhoul et al. 1999; Olson and Delen 2008]. There is a great number of metrics for classification evaluation, and although some are equivalent, most of them measure different aspects of the classification. When using metrics to evaluate a traffic classification mechanism, it is important to understand what is measured by each of them. In the following paragraphs, we briefly explain the most common metrics in traffic classification studies.

The accuracy of a classifier is usually evaluated by measuring its capability to correctly identify positive and negative cases. Hence, *accuracy* is defined as the ratio of correct positive and negative classifications to all the positive and negative cases in the experimental data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

However, accuracy is insufficient to evaluate a classifier when using imbalanced datasets with a greater number of positive or negative cases in the dataset, as it gives more importance to the most popular class in the dataset. In such cases, if a classifier privileges the class with more cases in the dataset, it will always achieve a good accuracy. For example, in an extreme case, a completely useless classifier that classifies as positive for every case in the dataset will achieve a high accuracy in a dataset containing, for example, 90% of positive cases. Therefore, it is necessary to use more than one measure, each of them evaluating different aspects of the results.

Two of those metrics, *precision* and *recall*, are used together to evaluate classification methods and are defined as follows [Nguyen and Armitage 2008b].

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}. \tag{3}$$

Some authors also used the term accuracy to refer to precision [Callado et al. 2010] or to recall [Hu et al. 2008]. These metrics are used to evaluate the capability of the classifier to correctly identify the positive cases. Precision, also referred to as *positive predictive value*, evaluates how correct the cases identified as positive by the classifier are, whereas recall, also referred to as *hit rate* or *true positive rate*, expresses the percentage of positive cases included in the dataset that were correctly identified by the classifier.

Nonetheless, precision and recall also have limitations in specific contexts, as they do not value rarity [Weiss 2004; Stefanowski and Wilk 2009]. Both metrics do not consider the amount of negative cases correctly identified by a classifier. This means that if a classifier *C1* returns, for example, ten false positives out of ten negatives and a classifier *C2* returns an equal number of false negatives and of true positives and ten false positives out of 1,000 negatives, both classifiers will have the same precision and recall. However, *C2* may be considered to have better performance, as it failed to correctly identify only 1% of the negative cases, while *C1* was not able to identify any negative case. Furthermore, the precision obtained for a dataset containing an extremely low share of positive cases may be affected by the high prevalence of negative cases. In fact, in such contexts, a very small percentage of the negative cases misclassified as positive cases may be sufficient to overcome the number of true positives identified by the classifier, due to the shortage of positive cases in the dataset.

In these situations, it may be advantageous to consider metrics that separately evaluate the classification of positive and of negative cases. Therefore, recall can be used together with a another metric, *specificity*, which is defined as follows [Wang 2008].

$$Specificity = \frac{TN}{FP + TN} \quad . \tag{4}$$

When used together with specificity, recall is usually called *sensitivity* [Raahemi et al. 2008b]. Sensitivity measures the ratio of correctly classified positive cases to the total of positive cases, whereas specificity evaluates the negative cases that were correctly classified. In the context of traffic classification, sensitivity and specificity are especially useful for evaluating classifiers that are focused on a specific class that accounts for a minority of the traffic in a dataset, for example, a classifier designed to identify video streaming or Voice over Internet Protocol (VoIP) traffic.

Moreover, Karagiannis et al. [2005a] defined a different metric similar to recall, which they called *completeness*, and used it together with precision, which they called *accuracy*. To the best of our knowledge, the two metrics were also used by Callado et al. [2009, 2010] and Szabó et al. [2007]. Completeness measures the ratio of classified positive cases, correctly or incorrectly, to the total number of positive cases and is defined as follows.

$$Completeness = \frac{TP + FP}{TP + FN}. \tag{5}$$

The metrics used to evaluate a classifier should be chosen depending on the context and purpose of each classifier. Although some authors have used different names for

similar metrics, in the next section, we will use the terms *accuracy*, *precision*, *recall*, *sensitivity*, *specificity*, and *completeness* as just described, so as to keep the article coherent.

5. DISCUSSION OF THE STATE OF THE ART ON TRAFFIC CLASSIFICATION

In the literature on traffic classification, several mechanisms and applications are proposed for the identification of application-level protocols. The following sections provide a theoretical study of the most relevant works in this field of study. Traffic classification methods are, in many cases, suitable for the identification of traffic from different types of applications. Nonetheless, since most of the applications whose traffic is difficult to identify by conventional means use P2P platforms, many of the studies discussed herein are oriented for the detection and classification of P2P traffic. Although the approaches used by most of the studies described in this section are, in many cases, also used for the detection of traffic anomalies, virus, and other software threats [Lakhina et al. 2005; Ranjan et al. 2007; Soewito et al. 2009], this section will be focused only on the studies addressing the subject of traffic classification.

5.1. Port-Based Classification

As described in Section 4.1, the early strategies for traffic classification were based on the identification of port numbers. The Internet Assigned Numbers Authority (IANA) keeps an updated list of the well-known or registered port numbers, which is available on the Web [IANA 2011]. Nevertheless, there are also port numbers or ranges that are traditionally used by some P2P systems. Table II presents a list of the port numbers commonly used by well-known P2P applications.

A few studies have used this approach to identify application protocols. Sen and Wang [2004] and Krishnamurthy and Wang [2002] analyzed P2P traffic collected at the border routers of a large ISP. In order to distinguish the flows from *Gnutella*, *FastTrack*, and *Direct Connect*, they used the TCP port numbers. Saroiu et al. [2002a] collected traffic from the University of Washington and, using port numbers, identified and analyzed the data from four content delivery systems: HTTP Web traffic, *Akamai* network, *KaZaA*, and *Gnutella*. Leibowitz et al. [2002] monitored traffic from an ISP network and analyzed *FastTrack*-based traffic, which includes *KaZaA*, *Morpheus*, and *Grokster* data, filtered through the use of port numbers. Gerber et al. [2003] resorted to port numbers as well to identify traffic from several P2P systems. They collected traffic from an ISP backbone and from a university network and analyzed its properties.

There are also tools for traffic analysis that provide information about the application-level protocol based on port numbers, like the *CoralReef* suite [Moore et al. 2001] or the *Wireshark* [2010] packet analyzer. A few studies have used the application port tables from *CoralReef* to identify the network traffic [Fraleigh et al. 2003].

5.2. Deep Packet Inspection Classification

The lack of effectiveness of the methods based on port numbers motivated an increase in the studies that analyze traffic using payload inspection. Sen et al. [2004] proposed payload signatures for *Gnutella*, *eDonkey*, *Direct Connect*, *BitTorrent*, and *KaZaA* and implemented them using the *Gigascope* monitor. They tested the solution using traffic collected on an access network to a major backbone and on a T3 (45 Mbps) link connecting a virtual private network (VPN) to the Internet. The authors estimated that the false positive rate was approximately 0%, while the false negative rate was between 0.00% and 4.97% for the analyzed protocols, with the exception of *BitTorrent*, for which it was 9.90%. However, they considered that the flows that use well-known port numbers of P2P applications are, in fact, P2P traffic. Based on that assumption, each flow

Table II. Well-known Port Numbers Used by Several P2P Protocols

Protocols	TCP Ports	UDP Ports
AIM - messages	5190	5190
AIM - video	1024–5000	1024–5000
ARES Galaxy	32285	32285
BitTorrent	6881–6999	
Blubster	41170–41350	41170–41350
Direct Connect	411, 412, 1025–32000	1025–32000
eDonkey	2323, 3306, 4242, 4500, 4501, 4661–4674, 4677, 4678, 4711, 4712, 7778	4665, 4672
FastTrack	1214, 1215, 1331, 1337, 1683, 4329	
Gnutella	6346, 6347	6346, 6347
GoBoogy	5335	5335
HotLine	5500–5503	
ICQ	5190	
iMesh	80, 443, 1863, 4329	
IRC	6665–6669	
Kazaa	1214	1214
MP2P	10240–20480, 22321, 41170	41170
MSN	1863	
MSN - file transfer	6891–6900	
MSN - voice	6901	6901
Napster	5555, 6666, 6677, 6688, 6699–6701, 6257	
PeerEnabler	3531	3531
Qnext	5235–5237	5235–5237
ROMnet	6574	
Scour Exchange	8311	
ShareShare	6399	6388, 6733, 6777
Soribada	7675–7677, 22322	7674, 22321
SoulSeek	2234, 5534	2234, 5534
WASTE	1337	1337
WinMX	6699	6257
XMPP/Jabber	5222, 5269	5222, 5269
Yahoo - messages	5050	
Yahoo - video	5100	
Yahoo - Voice	5000–5001	5000–5010

that used one of those ports and was not classified as P2P traffic was identified as a false negative case.

Moore and Papagiannaki [2005] presented a flow-based methodology that resorts to the deep inspection of the payloads. It uses a set of distinct methods that searches for known signatures within the full payload of each packet. The methods are checked sequentially until one of them matches a certain application. In the tests performed by the authors which relied on manual verification, the proposed set of methods was able to accurately identify approximately 99.99% of the traffic, which corresponds to the recall rate.

Karagiannis et al. [2004b] used payload signatures to identify traffic of several P2P applications, namely, *eDonkey2000*, *FastTrack*, *BitTorrent*, *WinMX*, *Gnutella*, *MP2P*, *Soulseek*, and *Direct Connect*. They used their approach to analyze a few traffic traces captured from links of two backbones, and conclude on the evolution of the percentage of P2P traffic in the Internet.

Spognardi et al. [2005] collected and analyzed traffic from *OpenNap*, *WPN*, and *FastTrack* P2P protocols in order to identify payload signatures. The signatures were codified in *form rules* for the *Snort* NIDS and used to monitor network traffic.

Choi and Choi [2006] proposed the use of port numbers as a real-time method for identifying the traffic. Afterwards, the traffic is also inspected offline using DPI techniques. The authors presented a methodology for checking if the packets match a data pattern that is based on an edit distance algorithm. Bin et al. [2007] proposed a solution that uses payload signatures to identify P2P flows as well. Each successfully identified packet is added to a table with a hash identifier, which is calculated from the source and destination IP addresses and from the transport port numbers. This way, the authors only examine the contents of the packets that belong to flows that were not classified yet.

The detection of chat-related traffic was studied in Dewes et al. [2003]. The authors analyzed several chat protocols and identified payload signatures. The tests show that the methodology presented, which was validated using manual verification, failed to detect less than 8.3% of all chat connections (recall of 91.7%), and from the ones detected, 93.13% were correctly classified (precision).

Generally, one the major drawbacks of DPI methods is their weight in terms of computation power. Hence, a few studies have tried to develop DPI mechanisms that are light and scalable. Risso et al. [2008] presented a taxonomy of the possible DPI approaches and performed a comparison of the performance and accuracy between a lightweight and a completely stateful traffic classification method. They concluded that, although the lightweight methods are not so accurate, they are still effective enough for the purpose of traffic classification while being able to perform much faster than the stateful approaches. Guo and Qiu [2008] proposed a signature-based method to identify P2P flows in high-speed networks using packet sampling and tested it with *BitTorrent*-related traffic. They evaluated the relation between its performance and the sampling probability, achieving different false positive and negative rates depending on the value of the sampling probability, from 0.00% to 11% and from 0.33% to 10.5%, respectively. In Cascarano et al. [2009], evaluated the computational cost of a DPI mechanism by comparing it with a statistical one. Although the comparison has been made between only two specific methods, it shows that, depending on the composition of the traces, the DPI mechanism can be as much computationally heavy as the statistical classifier; or it can go as high as five times the complexity of the statistical approach. In her Ph.D. dissertation, Yu [2006] developed high-speed packet processing algorithms, proposing the use of hardware support to perform the deep inspection of packets. Smith et al. [2008] used auxiliary variables and optimizations to implement a mechanism for deflating explosive deterministic finite automata (DFA). Using their solution, the authors were able to optimize the process of signature matching, achieving promising results for File Transfer Protocol (FTP), SMTP, and HTTP traffic. Kumar et al. [2006] introduced a new representation for regular expressions, called the delayed input DFA (D^2FA), which significantly reduces the space requirements of a DFA. The results of their tests showed that they were able to reduce memory space requirements by more than 95%. Cascarano et al. [2010a] presented two optimizations of a DPI classifier that reduce the data checked by the pattern matching engine. The improvements are achieved at the cost of a controlled reduction of the accuracy, which, unlike the case of intrusion detection, is acceptable in traffic classification.

The encryption of the payload is usually a problem for the DPI techniques. However, a few studies used the payload examination to identify P2P encrypted traffic. Carvalho et al. [2009b] manually identified several payload signatures of *BitTorrent*-encrypted traffic and provided a set of *Snort* rules to match the patterns observed. They tested the rules with traffic from a university network. The same authors have used a similar

approach to identify signatures for encrypted *eDonkey* traffic [Freire et al. 2009] and P2P TV traffic [Carvalho et al. 2009a].

Most DPI mechanisms are based on signature matching. Nevertheless, a few methods use the payload data in a different perspective. Dhamankar and King [2007] used entropy to explore the randomness of the data within the encrypted payloads of *Skype* traffic, resorting to clustering methods and congregating several heuristics. More studies have also addressed the subject of *Skype* traffic identification. Ehlert and Petgang [2006] described a detailed analysis of the *Skype* protocol and presented a signature for detecting its traffic based on payload and transport-level data.

Some authors have been developing studies on the automatic identification of payload signatures. Most of those studies are focused on the identification of worms, virus, and other traffic anomalies [Singh et al. 2004; Yegneswaran et al. 2005; Cavallaro et al. 2008]. However, a few authors have proposed similar approaches for traffic classification. Haffner et al. [2005] used three ML algorithmsm, and with two of them, they were able to construct signatures with precision between 99% and 100% and recall between 86.6% and 99.9% by resorting to the examination of a small amount of data at the beginning of the communication. The study was performed for traffic from FTP, SMTP, Post Office Protocol (POP), Internet Message Access Protocol (IMAP), HTTP, Hypertext Transfer Protocol Secure (HTTPS), and Secure Shell (SSH). Finamore et al. [2009] presented *KISS*, a classifier that automatically extracts signatures from a UDP stream by using a stochastic test that allows for the identification of the application protocol syntax, while ignoring the synchronization and semantic rules. The signatures can be seen as statistical fingerprints in the payload data. The authors tested the mechanism, verifying it manually using traffic traces from an Italian ISP. *KISS* correctly identified more than 98.1% of the samples in the worst case, reaching an average recall of 99.6% and an average false positive rate of 0.34%. Mantia et al. [2010] extended the previous method to also support the classification of TCP traffic, with an average recall of 97.62%. Park et al. [2008] also presented a solution for the automated creation of signatures, the *LASER* algorithm. The authors tested the approach for *LimeWire*, *BitTorrent*, and *Fileguri* using data collected in a campus network and manually verified. They achieved an accuracy rate of 97.39%, with a false negative rate between 0.39% and 10.40% and with 0% false positives.

5.3. Classification *In The Dark*

Recently, several studies have proposed classification strategies that rely on behavioral and statistical patterns, which can be further categorized as follows.

5.3.1. Heuristics. Several studies propose heuristics as a means to identify P2P traffic. Constantinou and Mavrommatis [2006] proposed a classifier that uses three heuristics: the number of hosts that act both as server and client in a specific port exceeds a given threshold; the estimated network diameter is at least as great as 2; and the number of hosts that are present in the first and last levels of the network exceeds a given threshold. The method was tested using data traces from *NLANR* [2010] and compared with port-based classification. Depending on the threshold values, the results vary between 8.5% and 12.7% of false negatives (detected with port-based and not detected with heuristics) and between 7.6% and 42.4% of additional positives (not detected with port-based and detected with heuristics). Perényi et al. [2006] described a method based on a set of six heuristics for identifying P2P traffic: simultaneous usage of TCP and UDP; the existence of several consecutive connections between two hosts; well-known P2P port numbers; multiple flows with the same flow identities; an IP using the same transport port more than five times in the measurement period; and the flow size larger than 1MB or its duration longer than ten minutes. The validation of

the approach was made using a small labeled traffic trace, and it achieved a recall of 99.14% and of 97.19% for P2P and non-P2P traffic, respectively, with 0.3% false positives and 0.8% false negatives. John and Tafvelin [2008] also proposed a set of heuristics to classify Internet traffic, which are a redefined combination of previously suggested ones [Karagiannis et al. 2004c; Perényi et al. 2006]: the concurrent use of TCP and UDP; the well-known P2P port numbers; the port numbers that are used very often; the relation between the number of IP addresses and the number of transport ports; and the flows carrying more than 1 MB or lasting more than ten minutes. Besides the heuristics, the authors also described a set of rules to reduce the number of false positive cases. They used the mechanism to classify traces collected at a university link, leaving only 2% of the traffic unclassified (recall of 98%).

5.3.2. Social Behavior. Karagiannis et al. [2005a] presented *BLINC*, a mechanism for flow classification that does not rely on the payload data or transport port numbers to identify the application protocol. *BLINC* analyzes traffic at three levels (social, functional, and application), exploiting properties of each node, like the relation with the remaining hosts, the role in the connection (server or client), the transport layer information, or the average packet size. The mechanism was tested using traffic collected at numerous academic, research, and residential complexes within a university campus, and it was evaluated by comparing it with a DPI-based method. *BLINC* was able to classify between 80% and 90% of the flows, corresponding to the completeness rate, with a precision ranging from 90% to 95%. Iliofotou et al. [2007] introduced a different perspective for the traffic analysis that is focused on the network-wide interactions of hosts. They model the social behavior of hosts by organizing and correlating the information in graphs, which they call traffic dispersion graphs (TDGs), where the edges represent different interactions. Iliofotou et al. [2008, 2009] used TDGs to create a framework, *Graption* (graph-based classification), to classify the traffic based on the application protocol. The mechanism was tested using two traces from a Tier-1 ISP and one trace from the *Abilene* (*Internet2*) network and a DPI-based method as baseline. The results showed that the solution was able to classify the traffic with a recall between 94% and 95% and a precision between 95% and 96%.

5.3.3. Statistical or Behavioral Signatures. A mechanism for flow classification based on the definition of statistical signatures or fingerprints for different traffic classes was proposed by Crotti et al. [2006, 2007]. The fingerprints are created using traffic preclassified with any effective mechanism and then used to classify network traffic. Dusi et al. [2008, 2009] also used statistical fingerprints to identify encrypted tunnels. The method was evaluated using data collected on controlled sessions and reaching a recall of between 82.45% and 100.00%. Bartlett et al. [2007a] identified three basic behavioral signatures from P2P file sharing: failed connections, the ratio of incoming and outgoing connections, and the use of unprivileged ports. They evaluated the mechanism by classifying *BitTorrent* and *Gnutella* traffic, captured from a commercial ISP and from academic institutions. In order to access the ground truth for *BitTorrent* data, the authors identified all the flows that used the default port number of *BitTorrent* tracker and manually verified that the destination was a real tracker. All the traffic identified by these means was confirmed to be P2P traffic. Additionally, they considered all flows using nonprivilege ports that are not well-known ports as *likely non-P2P*. For the *Gnutella* data, the authors considered as P2P all the flows that contact with *Gnutella* ultra-peers, which they identified by connecting repeatedly to the *Gnutella* network and keeping a record of the ultra-peers list. These approaches were used to verify the classification of P2P hosts and of *likely non-P2P* hosts. Besides this strategies, to verify the classification of the remaining flows, the authors identified the flows using default *BitTorrent* ports (6969, 6881–6888) and the default Gnutella port (6346). The

results show that *BitTorrent* hosts were detected with a recall ranging from 83% to 92%, while *Gnutella* hosts achieved a recall from 57% to 97%, and the false positive rate was between 2% and 25%. Freire et al. [2008a, 2008b] proposed a mechanism to identify VoIP calls hidden in Web traffic. The authors analyzed several properties of the network data to distinguish between VoIP and legitimate Web traffic and selected the following parameters: Web request size, Web response size, interarrival time between requests, number of requests per page, and page retrieval time. In order to measure the *goodness of fit*, they used the *Chi-square* and *Kolmogorov-Smirnov* tests. The evaluation was made using *Skype* and *Google Talk* VoIP data previously collected in both ISP and university links in a controlled way. The method achieved similar results for both protocols, being able to identify around 90% (recall rate) of the VoIP calls with a false positive rate of 2%, and 100% (recall rate) of VoIP calls hidden in Web traffic with a false positive rate of 5%. Gomes et al. [2008] analyzed several edge user traces of P2P and non-P2P applications and tried to identify a behavioral pattern of the P2P traffic. They concluded that the packet sizes of P2P traffic presented a high heterogeneity when compared to the packet sizes of the non-P2P traffic. They used entropy to represent the heterogeneity degree and calculated its value for a sliding window containing a fixed number of packets. P2P traffic, especially the one related with VoIP services, returned high entropy values, while for the regular client-server traffic, the entropy value was consistently smaller. Lin et al. [2009] proposed the use of packet-size distribution and port association as a pattern to distinguish application protocols. They used traces collected in a controlled environment to evaluate the method, which achieved a recall between 74% and 100% and false positive and negative rates ranging from 0% to 9% and from 0% to 18%, respectively. Palmieri and Fiore [2009] presented a new approach for the classification of Internet traffic that relies on recurrence quantification analysis (RQA). They studied nonlinear properties of specific IP flow types so that they could determine the recurrence phenomena and hidden nonstationary transition patterns related to each type of traffic. For the different traffic classes considered in the study (HTTP, *eDonkey2000*, domain name system (DNS), SMTP, POP, and SSH), the authors obtained average recall rates ranging from 45.8% to 89.4%, when compared to DPI. The tests were performed with three distinct traces captured in a university network.

5.3.4. Machine Learning Algorithms. The supervised and unsupervised ML methods are widely used in studies on the classification of network traffic. In the following paragraphs, the different research works based on ML are organized depending on the techniques employed.

Naïve Bayes and Neural Networks. A Naïve Bayes estimator was employed [Moore and Zuev 2005; Zuev and Moore 2005] to distinguish the traffic based on the application-level protocol, and hand-classified data was used to train the classifier. The input discriminators for this study were formed by several properties of the flows. The method was tested with traffic data from a research campus, previously hand-classified and collected twelve months later than the data used for the training process (which proves the temporal stability of this approach), and achieved a precision between 13.46% and 99.27% and a recall between 93.73% and 96.29%. Schmidt and Soysal [2006] proposed a mechanism for the detection of P2P traffic resorting to a Bayesian network, built using the following flow characteristics: IP packet size distribution, packets per flow distribution, octets per flow distribution, flow time distribution, and well-known port numbers. They evaluated the performance of the classifier using traffic from an academic network and compared the results against a signature-based method. The results showed false negative and positive rates ranging from 16% to 26% and from 22% to 28%, respectively. Auld et al. [2007] also described a classifier based on a Bayesian neural network, trained using data previously classified using DPI. A set of

traffic properties and statistics was used as input for the classification process. The method proved to have an accuracy between 95% and 99% for data manually verified and collected eight months after the data used to train the classification mechanism.

Clustering. McGregor et al. [2004] proposed a clustering-based methodology that extracts a range of flow properties and uses the expectation-maximization (EM) algorithm to cluster the flow into different classes. A preliminary validation of the approach showed promising results. A framework for traffic classification, based also on the EM algorithm and trained using several flow characteristics, was described in Zander et al. [2005a, 2005b]. The method was tested using traffic traces from *NLANR* [2010], and the results showed moderate effectiveness. Nguyen and Armitage [2006, 2008a] proposed a solution based on the EM algorithm and on a Naïve Bayes classifier. In order to test and classify the method, they used traffic from a gaming server and from a university link and obtained its ground truth using the port numbers. The results showed an average accuracy above 98.3%. Bernaille et al. [2006a; 2006b] presented a method for traffic classification that is based on the first five packets of a TCP connection, excluding the control packets (the ones marked with the flags *SYN*, *ACK*, etc.). They experimented with three clustering techniques to explore the relations between the initial packets and to identify clusters related with distinct application protocols: *k-means*, Gaussian mixture model (GMM), and spectral clustering. The mechanism was trained using a one-hour trace collected at the edge of a university network and was tested with a similar trace captured six months later, by comparing it with a DPI-based classifier. The results presented a recall between 36.0% and 100.0% and a false positive rate from 0.0% to 3.6%. Bernaille and Teixeira [2007] extended the same approach, using GMM to identify traffic encrypted (or tunneled) in secure sockets layer (SSL) connections. The evaluation, performed using manually generated traffic traces, showed a recall ranging from 81.20% to 100.00% and a false positive rate between 0.00% and 2.30%. Erman et al. [2006a] also described a preliminary work on the effectiveness of clustering algorithms for traffic classification. They employed the *k-means* and the density-based spatial clustering of applications with noise (DBSCAN) algorithms and used several properties to discriminate the flows, like the total number of packets, the mean packet size, the mean payload size, the number of bytes transfered, and the mean inter arrival time. The approach was tested using a publicly available trace without the payload data and a trace collected by the authors at a university link, showing a recall ranging from 86.6% to 93.5%. The ground truth verification was made using port numbers and DPI. Erman et al. [2006b] proposed an unsupervised ML solution—the EM algorithm—for the traffic classification. They analyzed the performance of the method using traffic traces collected at a university link and compared the results with a supervised ML technique, a Naïve Bayes classifier. The evaluation showed that the EM algorithm achieved precision and recall rates between 80% and 100%. Erman et al. [2007a] also proposed a semisupervised learning method for traffic discrimination, based on several flow-related statistics, that allows the classifiers to be designed from training data formed by a few labeled and many unlabeled flows. Although the mechanism is not limited to any specific clustering algorithm, after the previous studies, they decided to use *k-means*. They tested the mechanism using data whose ground truth was verified using DPI, heuristics, and manual verification and achieved a recall between 80% and 90%. The same approach was also used in Erman et al. [2007b] to distinguish between Web and P2P traffic, with an accuracy between 80% and 95%, precision between 71.42% and 97.84%, and recall between 62.10% and 97.32%.

Decision Trees. A method for traffic classification based on decision trees was proposed in Early et al. [2003]. The trees were constructed by employing the *C5.0* algorithm and using the information about the TCP flags used in each connection, as the authors

believed it to be enough to capture the flow behavior. The authors used HTTP, FTP, *Telnet*, SMTP, and SSH traffic to test and evaluate the mechanism, which proved to have a recall between 82% and 100%. Cao et al. [2008] described an approach for the identification of application protocols in real-time, at both host and flow levels, using a classification and regression tree (CART). Through offline analysis, they extracted metrics to characterize the traffic and used decision trees to identify the traffic in an online manner. The authors focused their experiments on traffic from *BitTorrent*, HTTP, SMTP, and FTP collected in a home link and also in an enterprise network. In order to assess the ground truth, the authors created the traces of *BitTorrent* actively in a controlled manner at a home environment. The HTTP, SMTP, and FTP traffic was captured in an enterprise network and filtered using the port numbers. In the tests the authors performed, the method classified the traffic with a false positive rate between 0.05% and 12.7% and a false negative rate between 0% and 17.9%. Raahemi et al. [2008b] applied concept-adapting very fast decision tree (CVFDT) to identify P2P traffic, using a set of network-level attributes of the packets. They used labeled datasets to evaluate the performance of the mechanism, achieving an accuracy between 79.50% and 98.65%, a specificity between 82.96% and 95.89%, and a specificity between 67.96% and 99.72%. Angevine and Zincir-Heywood [2008] used *C4.5* and the *AdaBoost* algorithms to classify UDP and TCP Skype flows. The authors used the mechanism to analyze labeled traffic traces from a university network with a recall between 94% and 99% and a false positive rate between 1% and 26%. A decision tree-based classifier, *Random Forests*, was used in Wang et al. [2008] to identify traffic from multiple P2P protocols. The method was tested with manually labeled datasets captured at residential and academic networks and achieved an accuracy rate ranging from 89.38% to 99.98%, a precision from 32.69% to 100.00%, and a false positive rate from 0.00% to 12.61%. Branch et al. [2009] also employed the *C4.5* algorithm using different conjunctions of flow features from packet lengths, statistics of large packets, and interarrival times. Using traffic from a university network, the method was able to classify the traffic with a precision of 99% and a recall of 98%.

Markov Chains and Models. Wright et al. [2006] focused specifically on the behavior of encrypted traffic. Using a classifier based on hidden Markov models and also on the *k-nearest neighbor* algorithm, they proved that it is possible to identify the application-level protocol in aggregate traffic without demultiplexing or reassembling the TCP connections; in aggregate traffic by demultiplexing the flows and analyzing them individually; and in aggregate traffic without demultiplexing the flows or recognizing which packets in the aggregate traffic belong to which flows (as when the traffic is encrypted at the network layer). The evaluation was performed using traffic from SMTP, HTTP, HTTPS, FTP, SSH, *Telnet*, and AOL Instant Messenger (AIM), and the ground truth information was verified using port numbers, presenting a recall ranging from 57.70% to 96.70% and a false positive rate between 0.62% and 8.37%. Dainotti et al. [2008] have also proposed a classification mechanism based on hidden Markov models, whose classification process is based on packet sizes and interpacket times. The authors applied the model to real traffic traces, verified manually and using DPI, of two multiplayer games, HTTP, SMTP, *eDonkey*, *PPlive* P2P TV, and *MSN Messenger*, reaching a recall of between 90.23% and 100.00%. Xusheng and Zhiming [2009] used Markov chains to model the sequences of control packets a certain application exchanges with a remote host and based the decision rule on the *Neyman-Pearson* test and on the likelihood criterion.

Support Vector Machine. Behavioral-based classification was accomplished in Gonzá1ez-Castaño et al. [2006] by employing SVMs. The solution proposed was evaluated using datasets that were labeled based on the port numbers and on a few

simple heuristics, reaching an accuracy between 78.7% and 90.2%. Turkett et al. [2008] extracted several flow features and used FTP-, SSH-, *Telnet*-, SMTP-, HTTP-, and POP-related traffic to train an SVM mechanism, which performed well in the tests conducted. Este et al. [2008] proposed three pattern recognition solutions based on SVMs, GMM, and *C4.5* to identify the presence of the unknown classes and used the size of the first packets as input feature. The tests performed with the three methods presented an accuracy between 92.53% and 98.83%, confirmed using DPI and manual verification. Este et al. [2009] carefully described the approach based on SVMs and used it to classify three sets of traffic. The results of the test showed a recall ranging from 69.6% to 100.0%. Valenti et al. [2009] described a new approach to identifying the traffic from P2P-TV applications resorting to the number of packets exchanged between the peers during short time intervals and used SVMs to train the mechanism. The authors captured traffic in a large testbed and used it to test the method, which was able to correctly classify between 91.3% and 99.6% the data (recall rate), with only between 0.3% and 8.7% false positives. An approach relying on SVMs was also proposed in Sena and Belzarena [2009]. The authors used the size of the first N packets of each flow as a feature for traffic classification and trained the mechanism using data previously classified through DPI. They tested the method using traffic from the network of an Uruguayan ISP and achieved an accuracy ranging from 30% to 100%.

Other Studies Relying on Machine Learning Techniques. Liu et al. [2007] used the ratio between the amount of download and upload traffic, in each minute, as an identification pattern for each application and proposed a supervised ML algorithm to identify each distinct class. They tested the method with traffic from a few P2P applications, namely *Maze*, *BitTorrent*, *PPlive*, *eDonkey*, and *thunder*, which they collected on a testbed. The results showed an accuracy rate between 78.5% and 99.8%, depending on the protocol. Raahemi et al. [2008a] employed Fuzzy Predictive Adaptive Resonance Theory (ART), or Fuzzy *ARTMAP*, to identify P2P traffic. They used only data from the IP headers to build the Fuzzy *ARTMAP* neural networks. The experimental tests, using labeled datasets, showed that the classifier is able to perform with an accuracy ranging from 78% to 92%, a sensitivity from 68% to 90%, and a specificity from 85% to 96%. Huang et al. [2008] used a set of discriminators from which they identified patterns by resorting to an ML technique. In this work, the authors experimented with a few techniques, concluding that Bayesian network, partial decision tree (PART), and *C4.5* are the ones that performed best. The evaluation was made using traces collected at a university link, whose ground truth was accessed using payload signatures. The method showed a recall between 90.87% and 95.11%, depending on the ML technique used. Hu et al. [2008, 2009] proposed a novel method for the classification of P2P traffic that aims to build behavioral profiles for each application by using *association rule mining*. They choose five flow tuples, extract flow statistics, and correlate them using the *Apriori* algorithm. The approach, which was tested for *BitTorrent* and *PPLive* using on-campus traces, verified manually and through DPI, presented a recall ranging from 90.0% to 98.0% and a false positive rate between 0.2% and 5.0%. Williams et al. [2006] compared five ML algorithms for traffic flow classification. They argued that it is useful to analyze the algorithms in terms of computational efficiency rather than classification accuracy as, even though the accuracy between distinct algorithms may be similar, the computational efficiency can be considerably different. Based on their results, the authors concluded that *C4.5* algorithm was able to classify the flows faster then the remaining algorithms. A similar conclusion was reached in Soysal and Schmidt [2007], in which three solutions for P2P flows detection based on ML were compared.

5.3.5. Service Identification. Baldi et al. [2009] described a new approach for traffic classification that relies on the identification of the service that generates the traffic. They

defined *service* as a triple formed by the IP address of the server, the transport port at the server, and the transport protocol. The authors say that the method can be seen as a complement to reducing the computation and memory requirements of the existing solutions. Nevertheless, in the tests performed on the Internet link of a university campus, the mechanism was able to successfully classify 81% of the packets and 93% of the data (recall rate).

5.4. Classification Based on Active Mechanisms

Although active methods are especially suitable for network performance studies, they can also be used on traffic detection mechanisms. The *Napster* and *Gnutella* systems were analyzed Saroiu et al. [2002b, 2003] with the purpose of characterizing the population of end-user hosts. The authors created a crawler for each of the systems that gathered information regarding different properties, like bottleneck bandwidths, IP-level latencies, etc. As the goal of the study was to characterize both systems, the results presented are not focused on the classification accuracy but in the properties of the traffic.

Ohzahata et al. [2005] have also proposed an active approach to identify pure P2P traffic and applied their methodology to the *Winny* P2P file-sharing system. They developed a crawler to collect information of the IP addresses and transport ports of the hosts connected to the system and used it to identify further peers. The study provides results regarding the number of peers identified by the mechanism, but its accuracy was not evaluated.

5.5. Classification through the Combination of Approaches

A few studies propose solutions that combine different kinds of approaches for the classification of network traffic. Karagiannis et al. [2004c] proposed a cross-validation mechanism which uses port numbers, payload signatures, and behavioral patterns to identify traffic from *eDonkey*, *Fasttrack*, *BitTorrent*, *Gnutella*, *MP2P*, *Direct Connect*, and *Ares*. Besides presenting payload signatures for the said applications, the authors propose a non-payload-based method that uses two heuristics. The first heuristic identifies source-destination IP pairs that use both TCP and UDP. The second one says that when the number of distinct IP addresses connected to a destination IP is equal to the number of distinct ports used for the connections, the flows are likely to be related to P2P applications. Their behavioral mechanism identified more than 90% of the total P2P bytes and 99% of the P2P flows, which corresponds to the recall rate. The false positive rate, which was calculated by comparing the results of the payload mechanism with the results of the behavioral method, is approximately 8% to 12% of the total estimate of P2P traffic. Nevertheless, the authors argue that part of the false positives are, in fact, true positives that were not identified by the payload-based mechanism.

Dedinski et al. [2005] presented an architecture for the detection and control of P2P traffic. It makes use of active crawlers to collect information about the peers of a specific application and understand the topology of the correspondent overlay network. Alongside, the network-level properties of the traffic are also analyzed (either per-packet or interpacket) and used as a behavioral pattern, which the authors identify using wavelet analysis techniques. They performed a preliminary test of the architecture using only *eDonkey* and FTP traffic.

A framework to identify Internet applications using a neural networks-based method was proposed [Nogueira et al. 2007, 2009; Couto et al. 2008] relying on a previous identification obtained through any alternative technique. The authors also described a module for classifying the traffic using payload signatures that was employed in the training of the neural network. They tested the method for traffic from *BitTorrent*,

eMule, *Gnutella*, and HTTP, collected individually for each application, achieving a recall between 90% and 99%.

Bonfiglio et al. [2007] proposed a Naïve Bayes classifier based on two traffic characteristics, the message size and the average interpackets gap. They also implemented a classifier based on the packet payload that uses the *Chi-Square* test to identify *Skype* traffic by exploiting the randomness induced by the encryption of the payload. The authors tested their approach using traffic from an ISP and from a campus and compared its accuracy against a signatures-based method, reaching a false positive rate between 0.00% and 2.40% and a false negative rate between 2.96% and 29.98%.

A mathematical framework for unsupervised protocol inference was described Ma et al. [2006]. The authors introduced three methods for identifying different aspects of the communication of a certain protocol: product distributions of byte offsets, Markov models of byte transitions, and common substring graphs of message strings. They evaluated the mechanism using traces collected at different buildings of the university campus and verified manually. Depending on each traffic class in the different traces, the precision was between 68.81% and 100.0% for product distributions, between 0.0% and 100.0% for Markov models, and between 76.87% and 99.99% for common substring graphs. The recall was between 81.82% and 100.0%, 0.0% and 100.0%, and 48.76% and 100.0%, for product distributions, Markov models, and common substring graphs, respectively.

Szabó et al. [2007] presented an architecture that can be extended with modules for distinct traffic classification approaches. They analyzed the performance of the solution using traffic captured in the network of five mobile operators in Europe and Asia. The effectiveness of the classification was also evaluated using hand-classified data and traces captured in a controlled environment.

Adami et al. [2009] proposed a new algorithm to identify, in real time, the hosts in a network that are using *Skype* clients, that relies in payload signatures and statistical analysis. The algorithm is able to recognize the different types of *Skype* communication: direct calls, calls using a relay node, call to phone service, and file transfer. It was tested online and offline using traffic from a university LAN and from a small LAN connected to an ADSL link, and its performance was compared with the performance of five other classifiers. The traces used in Bonfiglio et al. [2007] were also tested. The results showed a percentage of false positives between 0% and 0.01% and of false negatives between 0.06% and 0.64%, in terms of bytes and flows and for both TCP and UDP traffic. The exception was the false negative rate for TCP flows, whose value was 27.46%.

Callado et al. [2010] collected five distinct datasets, captured in different contexts. The first one was formed by traces from individual applications captured in client computers, which where assembled in a single dataset. By creating the traces manually, the authors could be sure of the applications that generated the traffic. The other datasets were captured in a laboratory network, in an academic backbone, and in the core router of a commercial link (only one direction). The fifth dataset was formed by the traffic in only one direction of the trace from the academic backbone. The ground truth of these four datasets was obtained using DPI. The authors then used six ML algorithms implemented in *Weka* [Hall et al. 2009] to classify the traffic in the five datasets: *J48* (*C4.5* decision trees), PART, *NBTree* (decision trees with Naïve Bayes classifiers on the leaves), Bayesian networks with simple estimator, Bayesian networks with kernel estimator, and SVMs. They concluded that none of the classifiers performed better in all the datasets (which correspond to different contexts), and thus they presented a method to combine different classifiers. In order to choose the result of the combination, they proposed four algorithms: *random selection*, *maximum likelihood combination*, *Dempster-Shafer combination*, and *enhanced Dempster-Shafer combination*. Although it was tested mostly with ML algorithms, the method is independent of the classifiers

one may want to combine. In fact, they have also used *BLINC* [Karagiannis et al. 2005a] and DPI in some of the combinations. The results of the evaluation showed that the precision varies from 60% to 99% and the completeness varies from 90% to 100%, depending on the dataset used. Nevertheless, the lower accuracy values were obtained for the datasets that contained only one direction of the flows.

5.6. Applications for Traffic Classification

Besides the research studies proposing solutions for traffic classification and the commercial tools, there are also a few available and *ready to use* applications, which are described in the following paragraphs.

L7-filter [2010] is a classification tool for the Linux *Netfilter* subsystem that uses the application-layer data to identify the packets. It is widely used in many studies, being, most of the time, the comparison baseline for the performance evaluation of new methods. Another DPI-based tool is *l7-netpdlclassifier* [2010], which is based on the *NetBee* [2010] library and uses a signature database [NetPDL 2010] written using the *NetPDL* language [Risso and Baldi 2006]. *ipoque* has also made available an open-source version of their DPI tool, which they called *OpenDPI* [2010].

Antoniades et al. [2006] developed *Appmon* [2010], a tool for the application-level classification of network traffic. It is based on two *MAPI* [2010] function libraries and relies on port numbers and data signatures to identify the protocols.

Dainotti et al. [2009] presented *TIE* [2010], a novel community-oriented software for traffic classification. It uses the *libpcap* library and it supports distinct definitions of sessions and classes, as well as allowing for the implementation of additional classification plugins. In its initial state, *TIE* is available with only two classification plugins: port numbers (based on *CoralReef*) and payload signatures (based on *L7-filter*).

A classification tool based on clustering mechanisms, called *NetADHICT* [2010], is proposed in Inoue et al. [2007]. It decomposes the traffic without the use of any application-specific knowledge, and it uses an *AJAX*-based Web interface that allows one to see the high-level structure of network traffic.

Although *CoralReef*, *Snort*, and *Wireshark* are not especially intended to classify the traffic regarding the application protocol, they do provide simple mechanisms for such a purpose. The *CoralReef* suite gives the user the ability to identify the application-level protocol based on the port numbers; *Snort*, by default, contains several rules to identify signatures in the contents of the packets of several protocols; while *Wireshark* is also able to recognize payload patterns in non-encrypted traffic.

5.7. Summary and Challenges

Tables III, IV, and V summarize the textual analysis included in the previous sections. Furthermore, the chronological ordering of the tables allows one to observe the evolution of the approaches used for the traffic classification and of the protocols each study addressed. In the tables, the *Protocols* columns are related with the protocols considered by each study. The performance metrics included in these tables were chosen as they were the most used ones in the research works described along this survey.

The studies proposing DPI-based solutions are listed in Table III, along with the indication of which were used by their authors to identify encrypted or obfuscated traffic. For the sake of simplicity, the studies that evaluated or compared the performance of existing methods were not included in the table.

Since VoIP traffic raises special concerns for network administrators, a few recent studies have been presenting mechanisms for its detection. Table IV provides a summarized analysis of the approaches they used and their performance.

Table V describes the characteristics and performance of most of the studies analyzed in this survey that present methods for classification *in the dark*. For the sake of

Table III. Studies Based on DPI and Their Capability to be Applied to Encrypted Traffic

Studies	Protocols	Encryption
Dewes et al. [2003]	Chat protocols	Does not apply
Sen et al. [2004]	*Gnutella*, *eDonkey*, *Direct Connect*, *BitTorrent*, *KaZaA*	Does not apply
Karagiannis et al. [2004b]	*eDonkey2000*, *FastTrack*, *BitTorrent*, *WinMX*, *Gnutella*, *MP2P*, *Soulseek*, *Direct Connect*	Does not apply
Moore and Papagiannaki [2005]	multiple protocols	Does not apply
Spognardi et al. [2005]	*OpenNap*, *WPN*, *FastTrack*	Does not apply
Haffner et al. [2005]	FTP, SMTP, POP, IMAP, HTTP, HTTPS, SSH	Apply
Choi and Choi [2006]	multiple protocols	Does not apply
Ehlert and Petgang [2006]	*Skype*	Apply
Bin et al. [2007]	P2P	Does not apply
Dhamankar and King [2007]	multiple protocols	Apply
Guo and Qiu [2008]	*BitTorrent*	Does not apply
Smith et al. [2008]	FTP, SMTP, HTTP	Does not apply
Park et al. [2008]	*LimeWire*, *BitTorrent*, *Fileguri*	Apply
Carvalho et al. [2009a]	P2P TV	Apply
Carvalho et al. [2009b]	*BitTorrent*	Apply
Freire et al. [2009]	*eDonkey*	Apply
Finamore et al. [2009]	multiple protocols	Apply
Mantia et al. [2010]	multiple protocols	Apply
Cascarano et al. [2010a]	multiple protocols	Does not apply

Table IV. Studies Addressing the Subject of VoIP Traffic Identification and an Overview of Their Performance in Terms of Precision (P), Recall (R), False Positives (FP), or False Negatives (FN)

Studies	Approach	Protocols	Performance (%)
Bonfiglio et al. [2007]	Naïve Bayes and *Chi-Square* test	*Skype*	FP: 0.00–2.40; FN: 2.96–29.98
Angevine and Zincir-Heywood [2008]	*C4.5* and *AdaBoost*	*Skype*	R: 94–99; FP: 1–26
Freire et al. [2008a; 2008b]	behavioral signatures	*Skype* and *Google Talk*	R: 90–100; FP: 2–5
Branch et al. [2009]	*C4.5*	*Skype*	P: 99; R: 98
Adami et al. [2009]	DPI and statistical analysis	*Skype*	FP: 0–0.01; FN: 0.06–27.46

simplicity, only the studies that proposed a new method and evaluated its performance were included. The *Baseline* column indicates how the ground truth information of the traffic used in the evaluation was assessed, or what method was used as a reference to calculate the accuracy value. Most of the terms used in this column are easily understandable. The expression *manual* refers to traces manually verified or classified, *controlled traces* refers to manually or actively generated traces, and *testbed* refers to traces captured in previously prepared testbeds. This table is not meant to be a comparison between the methods, as the evaluations were made by the authors under different conditions and using distinct metrics [Salgarelli et al. 2007]. Its only purpose is to provide an overview of the behavioral methods presented in the literature.

Likewise, Table VI provides a side-by-side comparison of the different approaches followed by several studies in the literature. In order to keep the table short, we added only a maximum of four studies for each type of method and gave priority to the most recent ones. The evaluation results were included to give an easy perception of the performance of each method. Additionally, another column was added to describe the

Table V. Summary of the Studies Presenting New Methods for Traffic Classification *in the dark* and an Overview of Their Performance in Terms of Accuracy (A), Precision (P), Recall (R), Sensitivity (Sens), Specificity (Spec), Completeness (C), False Positives (FP), or False Negatives (FN)

Studies	Approach	Protocols	Performance (%)	Baseline
Early et al. [2003]	*C5.0*	HTTP, FTP, *Telnet*, SMTP, SSH	R: 82–100	ports
Karagiannis et al. [2004c]	DPI and heuristics	*eDonkey*, *Fasttrack*, *BitTorrent*, *Ares*, *Gnutella*, *MP2P*, *Direct Connect*	R: 90–99; FP: 8–12	DPI
Karagiannis et al. [2005a]	social behavior	multiple protocols	P: 95–99; C: 80–90	DPI
Moore and Zuev [2005]	Naïve Bayes	multiple protocols	P: 13.46–99.27; R: 93.73–96.29	manual
Constantinou and Mavrommatis [2006]	heuristics	P2P	FP: 7.6–42.4; FN: 8.5–12.7	ports
Wright et al. [2006]	hidden Markov models and *k-nearest neighbor*	SMTP, HTTP, HTTPS, FTP, SSH, *Telnet*, AIM	R: 57.70–96.70; FP: 0.62–8.37	ports
Schmidt and Soysal [2006]	Naïve Bayes	P2P	FP: 22–28; FN: 16–26	DPI
Gonzá1ez-Castaño et al. [2006]	SVMs	multiple protocols	A: 78.7–90.2	ports and heuristics
Perényi et al. [2006]	heuristics	P2P	R: 97.19–99.14; FP: 0.3; FN: 0.8	small labeled trace
Ma et al. [2006]	product dists, Markov models, and substring graphs	multiple protocols	P: 0.0–100.0; R: 0.0–100.0	manual
Bernaille et al. [2006a, 2006b]	*k-means*, GMM, and spectral clustering	multiple protocols	R: 36.0–100.0; FP: 0.0–3.6	DPI
Erman et al. [2006a, 2006b, 2007a, 2007b]	*k-means* and DBSCAN	multiple protocols	A: 80–95; P: 71.42–100.00; R: 62.10–100.00	ports and DPI
Nguyen and Armitage [2006, 2008a]	EM and Naïve Bayes	multiple protocols	P: 98.3–99.7; R: 96.0–98.9	ports
Bernaille and Teixeira [2007]	GMM	multiple protocols	R: 81.20–100.00; FP: 0.00-2.30	controlled traces
Auld et al. [2007]	Bayesian neural networks	multiple protocols	A: 95–99	manual
Liu et al. [2007]	supervised learning algorithm	*Maze*, *BitTorrent*, *PPlive*, *eDonkey*, *thunder*	A: 78.5–99.8	testbed
Bartlett et al. [2007a]	behavioral signatures	P2P file sharing	R: 57–97; FP: 2–25	ports and manual
Nogueira et al. [2007, 2009]	DPI and neural networks	*BitTorrent*, *eMule*, *Gnutella*, HTTP	R: 90–99	individual traces
John and Tafvelin [2008]	heuristics	multiple protocols	R: 98	comparison other methods
Wang et al. [2008]	*Random Forests*	multiple protocols	A: 89.38–99.98; P: 32.69–100.00; FP: 0.00–12.61	labeled traces
Cao et al. [2008]	CART	*BitTorrent*, HTTP, SMTP, FTP	FP: 0.05–12.7; FN: 0–17.9	ports and controlled traces

(Continued)

Table V. (Continued)

Studies	Approach	Protocols	Performance (%)	Baseline
Dainotti et al. [2008]	hidden Markov models	gaming, HTTP, SMTP, *eDonkey*, *PPlive*, *MSN*	R: 90.23–100.00	DPI and manual
Raahemi et al. [2008b]	CVFDT	P2P	A: 79.50–98.65; Sens: 82.96–95.89; Spec: 67.96–99.72	labeled traces
Raahemi et al. [2008a]	Fuzzy *ARTMAP* neural networks	P2P	A: 78–92; Sens: 68–90; Spec: 85–96	labeled traces
Huang et al. [2008]	Bayesian network, PART, and *C4.5*	multiple protocols	R: 90.87–95.11	DPI
Este et al. [2008, 2009]	SVMs, GMM, and *C4.5*	multiple protocols	A: 92.53–98.83; R: 69.6–100.0	DPI and manual
Iliofotou et al. [2008, 2009]	social behavior	P2P	P: 95–96; R: 94–95	DPI
Dusi et al. [2008, 2009]	behavioral signatures	encrypted tunnels	R: 82.45–100.00	testbed
Hu et al. [2008, 2009]	behavioral signatures	*BitTorrent* and *PPLive*	R: 90.0–98.0; FP: 0.2–5.0	DPI and manual
Lin et al. [2009]	behavioral signatures	multiple protocols	R: 74–100; FP: 0–9; FN: 0–18	testbed
Valenti et al. [2009]	SVMs	P2P-TV	R: 91.3–99.6; FP: 0.3–8.7	testbed
Sena and Belzarena [2009]	SVMs	multiple protocols	A: 30–100	DPI
Palmieri and Fiore [2009]	behavioral signatures	HTTP, SMTP, DNS, POP, SSH, *eDonkey*	R: 45.8–89.4	DPI
Baldi et al. [2009]	service-based	multiple protocols	R: 81–93	client probe
Callado et al. [2010]	combination of methods	multiple protocols	P: 60–99; C: 90–100	DPI and controled traces

ability of the method to be applied to traffic with encrypted transport-level payloads. Although the studies based on port numbers did not addressed the encryption issue, we considered them suitable for encrypted traffic, since the TCP and UDP port numbers are usually not encrypted.

The literature review presented in this section shows a clear trend towards the use of classification *in the dark* methods. The majority of the articles published in the last few years proposed alternatives to DPI that can be used for encrypted or obfuscated traffic and can operate in real-time in high-speed networks. This tendency is driven by the growth of the networks throughput and need to have means to identify the nature of the traffic, and also by the increasingly common payload encryption.

The early methods for traffic classification *in the dark* were mostly based on behavior modeling, either by resorting to heuristics or by implementing more complex mechanisms. More recently, however, most studies are proposing classifiers based on statistical signatures or in multiple ML algorithms. Although the number of proposals based on ML is growing significantly, they seem to have reached a point where most of them use similar ML algorithms to process different features of the traffic, and all of them present high accuracy. Hence, it is difficult to be sure if such proposals are evolving the state of the art in traffic classification. Some of the recent articles are still proposing methods implemented to work offline only, as they need to have access to the entire flows. Moreover, the methods for classification *in the dark* are growing in complexity, compromising one of their main motivations. In fact, Cascarano et al. [2009] compared the performance of a DPI classifier and an SVM-based method and

Table VI. Overview of Studies for Traffic Classification that Follow Different Approaches, Including Their Ability to be Applied to Encrypted Traffic and Their Performance in Terms of Accuracy (A), Precision (P), Recall (R), Sensitivity (Sens), Specificity (Spec), Completeness (C), False Positives (FP), or False Negatives (FN)

Appr.	Methods	Studies	Performance (%)	Encryption
Port based	port numbers identification	Saroiu et al. [2002a]	–	Apply
		Gerber et al. [2003]	–	Apply
		Fraleigh et al. [2003]	–	Apply
		Sen and Wang [2004]	–	Apply
DPI	payload strings	Sen et al. [2004]	FP: 0; FN: 0.00–9.90	Does not apply
		Moore and Papagiannaki [2005]	R: 99.99	Does not apply
		Guo and Qiu [2008]	FP: 0.00–11; FN: 0.33–0.5	Does not apply
		Cascarano et al. [2010a]	–	Does not apply
	automated signature extraction	Haffner et al. [2005]	P: 99.0–100; R: 86.6–99.9	Apply
		Park et al. [2008]	A: 97.39; FP: 0.39–10.40; FN: 0	Apply
		Finamore et al. [2009]	R: 99.6; FP: 0.34	Apply
		Mantia et al. [2010]	R: 97.62	Apply
	heuristics based on payload bytes	Ehlert and Petgang [2006]	–	Apply
	payload randomness	Dhamankar and King [2007]	–	Apply
	string matched using DFA	Smith et al. [2008]	–	Does not apply
Classification In The Dark	heuristics	Constantinou and Mavrommatis [2006]	FP: 7.6–42.4; FN: 8.5–12.7	Apply
		Perényi et al. [2006]	R: 97.19–99.14; FP: 0.3; FN: 0.8	Apply
		John and Tafvelin [2008]	R: 98	Apply
	social behavior	Karagiannis et al. [2005a]	P: 95–99; C: 80–90	Apply
		Iliofotou et al. [2008, 2009]	P: 95–96; R: 94–95	Apply
	statistical or behavioral signatures	Freire et al. [2008a, 2008b]	R: 90–100; FP: 2–5	Apply
		Dusi et al. [2008, 2009]	R: 82.45–100.00	Apply
		Lin et al. [2009]	R: 74–100; FP: 0–9; FN: 0–18	Apply
		Palmieri and Fiore [2009]	R: 45.8–89.4	Apply
	Naïve Bayes and neural networks	Moore and Zuev [2005]	P: 13.46–99.27; R: 93.73–96.29	Apply
		Schmidt and Soysal [2006]	FP: 22–28; FN: 16–26	Apply
		Auld et al. [2007]	A: 95–99	Apply
	clustering	Bernaille et al. [2006a, 2006b]	R: 36.0–100.0; FP: 0.0–3.6	Apply
		Erman et al. [2006a, 2006b, 2007a, 2007b]	A: 80–95; P: 71.42–100.00; R: 62.10–100.00	Apply
		Nguyen and Armitage [2006, 2008a]	P: 98.3–99.7; R: 96.0–98.9	Apply
		Bernaille and Teixeira [2007]	R: 81.20–100.00; FP: 0.00–2.30	Apply

(Continued)

Table VI. (Continued)

Appr.	Methods	Studies	Performance (%)	Encryption
		Early et al. [2003]	R: 82–100	Apply
		Cao et al. [2008]	FP: 0.05–12.7; FN: 0–17.9	Apply
	decision trees	Angevine and Zincir-Heywood [2008]	R: 94–99; FP: 1–26	Apply
		Branch et al. [2009]	P: 99; R: 98	Apply
	Markov chains and models	Wright et al. [2006]	R: 57.70–96.70; FP: 0.62–8.37	Apply
		Dainotti et al. [2008]	R: 90.23–100.00	Apply
		Gonzá1ez-Castaño et al. [2006]	A: 78.7–90.2	Apply
		Este et al. [2008, 2009]	A: 92.53–98.83; R: 69.6–100.0	Apply
	SVMs	Valenti et al. [2009]	R: 91.3–99.6; FP: 0.3–8.7	Apply
		Sena and Belzarena [2009]	A: 30–100	Apply
		Liu et al. [2007]	A: 78.5–99.8	Apply
		Raahemi et al. [2008a]	A: 78–92; Sens: 68–90; Spec: 85–96	Apply
	other ML-based methods	Huang et al. [2008]	R: 90.87–95.11	Apply
		Hu et al. [2008, 2009]	R: 90.0–98.0; FP: 0.2–5.0	Apply
	service identification	Baldi et al. [2009]	R: 81–93	Apply
Active Mechanisms	Active crawlers	Saroiu et al. [2002b, 2003]	–	Apply
		Ohzahata et al. [2005]	–	Apply
	DPI and heuristics	Karagiannis et al. [2004c]	R: 90–99; FP: 8–12	Apply
Combination of Approaches	product dists, Markov models, and substring graphs	Ma et al. [2006]	P: 0.0–100.0; R: 0.0–100.0	Does not apply
	DPI, heuristics, and ports	Szabó et al. [2007]	–	Does not apply
	Naïve Bayes and payload randomness	Bonfiglio et al. [2007]	FP: 0.00–2.40; FN: 2.96–29.9	Apply
	DPI and neural networks	Nogueira et al. [2007, 2009]	R: 90–99	Apply
	DPI and statistical analysis	Adami et al. [2009]	FP: 0.00–0.01; FN: 0.06–27.46	Apply
	combination of methods	Callado et al. [2010]	P: 60–99; R: 90–100	Apply

concluded that they have similar computation cost. On the other hand, some recent studies are also proposing DPI methods that are able to classify encrypted traffic (see Table III).

Therefore, more effort should be put on strategies to evaluate the true performance of the classifiers. This is not a simple task and it raises many challenges, as described in Section 4.5. However, it is crucial, not to compare classifiers, but to have an accurate perception if the current proposals are really effective and how they can be improved.

To the best of our knowledge, only three articles have addressed the subject of ground truth verification and proposed solutions [Szabó et al. 2008; Gringoli et al. 2009; Canini et al. 2009]. Moreover, a correct performance evaluation depends also on the datasets used for the validation. The classification challenges raised by several applications should be carefully analyzed and perhaps datasets of the traffic from many of them could be made available to be used in research studies.

Furthermore, there are several available tools, ready to use, for traffic classification using DPI; however, there are almost no applications implementing traffic characterization *in the dark* methods and that can be easily installed and experimented, online and offline. Although this is not a clear research goal, it would be interesting to be able to effortlessly use some of the proposed methods in real-time experimental network environments and see how they could adapt to real scenarios.

6. CONCLUSIONS

The evolution of the services and applications running on the Internet has caused important changes in the properties of the traffic. Besides the increase of bandwidth consumption, other challenges have been raised for network managers. In order to guarantee the correct operation of networks, efficient mechanisms for traffic classification are required. Since port-based methods have lost their utility when the protocols started to use random port numbers, many studies proposed alternative mechanisms to classify traffic, either by deeply inspecting the traffic or using behavioral information.

This article presents a survey on traffic classification that describes carefully the existing approaches. An extensive analysis of the literature was provided, pointing out the achievements and strengths of each study and its main goals. For the sake of understanding, it also included an introduction to the subject of traffic measuring for the purpose of network monitoring.

The analysis of the literature bespeaks a clear interest of researchers, in the last years, in traffic classification, motivated by the challenges created by new services and protocols, especially the ones based on P2P architecture. Furthermore, the evolution of the studies on this topic shows an increasing concern about the encryption of the traffic and its consequences for traffic management. The search for more accurate behavioral methods and DPI mechanisms capable of processing traffic in high-speed networks, together with the capability to classify encrypted traffic, seem to be strong trends for the future.

ACKNOWLEDGMENTS

The authors are thankful to all the anonymous reviewers for constructively criticizing this work.

REFERENCES

Adami, D., Callegari, C., Giordano, S., Pagano, M., and Pepe, T. 2009. A real-time algorithm for Skype traffic detection and classification. In *Proceedings of the 9th International Conference on Next Generation Wired/Wireless Networking (NEW2AN'09)*. Lecture Notes in Computer Science, vol. 5764. Springer-Verlag, Berlin Heidelberg, 168–179.

Allman, M. and Paxson, V. 2007. Issues and etiquette concerning use of shared measurement data. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'07)*. ACM, New York, NY, 135–140.

Amer, P. D. and Cassel, L. N. 1989. Management of sampled real-time network measurements. In *Proceedings of the 14th IEEE Conference on Local Computer Networks (LCN'89)*. IEEE Press, New York, NY, 62–68.

Angevine, D. and Zincir-Heywood, A. N. 2008. A preliminary investigation of Skype traffic classification using a minimalist feature set. In *Proceedings of the 3rd International Conference on Availability, Reliability and Security (ARES'08)*. IEEE Computer Society Press, 1075–1079.

Antoniades, D., Polychronakis, M., Antonatos, S., Markatos, E. P., Ubik, S., and Øslebø, A. 2006. Appmon: An application for accurate per application network traffic characterization. In *Proceedings of the IST Broadband Europe Conference.*

Apisdorf, J., Claffy, K. C., Thompson, K., and Wilder, R. 1996. OC3MON: Flexible, affordable, high performance statistics collection. In *Proceedings of the 10th USENIX Systems Administration Conference (LISA'96).* USENIX Association, Berkeley, CA, 97–112.

Appmon. 2010. Appmon description. `http://lobster.ics.forth.gr/~appmon/appmon_description.html`. (Last accessed 3/10).

Arlitt, M. and Williamson, C. 2007. The extensive challenges of Internet application measurement. *IEEE Netw. 21,* 3, 41–46.

Auld, T., Moore, A. W., and Gull, S. F. 2007. Bayesian neural networks for Internet traffic classification. *IEEE Trans. Neural Netw. 18,* 1, 223–239.

Azzouna, N. B. and Guillemin, F. 2003. Analysis of ADSL traffic on an IP backbone link. In *Proceedings of the IEEE Global Communications Conference (GlobeCom'03)*, Vol. 7. IEEE, 3742–3746.

Baldi, M., Baldini, A., Cascarano, N., and Risso, F. 2009. Service-based traffic classification: Principles and validation. In *Proceedings of the IEEE Sarnoff Symposium (SARNOFF'09)*. IEEE Press, Piscataway, NJ, 115–120.

Bartlett, G., Heidemann, J., and Papadopoulos, C. 2007a. Inherent behaviors for on-line detection of peer-to-peer file sharing. In *Proceedings of the IEEE Global Internet Symposium*. IEEE, 55–60.

Bartlett, G., Heidemann, J., and Papadopoulos, C. 2007b. Understanding passive and active service discovery. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'07)*. ACM, New York, NY, 57–70.

Basher, N., Mahanti, A., Mahanti, A., Williamson, C., and Arlitt, M. 2008. A comparative analysis of web and peer-to-peer traffic. In *Proceedings of the 17th International Conference on World Wide Web (WWW'08).* ACM, New York, NY, 287–296.

Bernaille, L. and Teixeira, R. 2007. Early recognition of encrypted applications. In *Proceedings of the Passive and Active Measurement Conference (PAM'07)*. Lecture Notes in Computer Science, vol. 4427, Springer-Verlag, Berlin Heidelberg, 165–175.

Bernaille, L., Teixeira, R., Akodjenou, I., Soule, A., and Salamatian, K. 2006a. Traffic classification on the fly. *ACM SIGCOMM Comput. Commun. Rev. 36,* 2, 23–26.

Bernaille, L., Teixeira, R., and Salamatian, K. 2006b. Early application identification. In *Proceedings of the 2nd Conference on Future Networking Technologies (CoNEXT'06)*. ACM, 1–12.

Bin, L., Zhi-Tang, L., and Hao, T. 2007. A methodology for P2P traffic measurement using application signature work-in-progress. In *Proceedings of the 2nd International Conference on Scalable Information Systems (InfoScale'07)*, vol. 304. ICST, Brussels, Belgium, 1–2.

Bonfiglio, D., Mellia, M., Meo, M., Rossi, D., and Tofanelli, P. 2007. Revealing Skype traffic: When randomness plays with you. *ACM SIGCOMM Comput. Commun. Rev. 37,* 4, 37–48.

Branch, P. A., Heyde, A., and Armitage, G. J. 2009. Rapid identification of Skype traffic flows. In *Proceedings of the 18th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'09)*. ACM, New York, NY, 91–96.

Bro. 2010. Bro intrusion detection system. `http://bro-ids.org`. (Last accessed 3/10).

Cáceres, R., Duffield, N., Feldmann, A., Friedmann, J. D., Greenberg, A., Greer, R., Johnson, T., Kalmanek, C. R., Krishnamurthy, B., Lavelle, D., Mishra, P. P., Rexford, J., Ramakrishnan, K. K., True, F. D., and van der Merwe, J. E. 2000. Measurement and analysis of IP network usage and behavior. *IEEE Commun. Mag. 38,* 5, 144–151.

Callado, A., Kamienski, C., Szabó, G., Gero, B. P., Kelner, J., Fernandes, S., and Sadok, D. 2009. A survey on Internet traffic identification. *IEEE Commun. Surveys Tuts. 11,* 3, 37–52.

Callado, A., Kelner, J., Sadok, D., Kamienski, C. A., and Fernandes, S. 2010. Better network traffic identification through the independent combination of techniques. *J. Netw. Comput. Appl. 33,* 4, 433–446.

Canini, M., Li, W., Moore, A. W., and Bolla, R. 2009. GTVS: Boosting the collection of application traffic ground truth. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA'09)*. Springer Verlag, Heidelberg, Germany, 54–63.

Cao, J., Chen, A., Widjaja, I., and Zhou, N. 2008. Online identification of applications using statistical behavior analysis. In *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom'08)*. IEEE, 1–6.

Carvalho, D. A., Pereira, M., and Freire, M. M. 2009a. Detection of peer-to-peer TV traffic through deep packet inspection. In *Acta da 9ª Conference sobre Redes de Computadores*. INESC-ID and Instituto Superior Técnico, 6.

Carvalho, D. A., Pereira, M., and Freire, M. M. 2009b. Towards the detection of encrypted BitTorrent traffic through deep packet inspection. In *Proceedings of the International Conference on Security Technology (SecTech'09)*. Communications in Computer and Information Science Series, vol. 58, Springer-Verlag, Berlin Heidelberg, 265–272.

Cascarano, N., Ciminiera, L., and Risso, F. 2010a. Improving cost and accuracy of DPI traffic classifiers. In *Proceedings of the 25th ACM Symposium on Applied Computing (SAC'10)*. ACM, New York, NY, 641–646.

Cascarano, N., Este, A., Gringoli, F., Risso, F., and Salgarelli, L. 2009. An experimental evaluation of the computational cost of a DPI traffic classifier. In *Proceedings of the IEEE Global Communications Conference (GlobeCom'09)*. IEEE, 1–8.

Cascarano, N., Risso, F., Este, A., Gringoli, F., Finamore, A., and Mellia, M. 2010b. Comparing P2PTV traffic classifiers. In *Proceedings of the IEEE International Conference on Communications (ICC'10)*. IEEE, 1–6.

Cavallaro, L., Lanzi, A., Mayer, L., and Monga, M. 2008. LISABETH: Automated content-based signature generator for zero-day polymorphic worms. In *Proceedings of the 4th International Workshop on Software Engineering for Secure Systems (SESS'08).* ACM, New York, NY, 41–48.

Choi, K. and Choi, J. K. 2006. Pattern matching of packet payload for network traffic classification. In *Proceedings of the Joint International Conference on Optical Internet and Next Generation Network (COIN-NGNCON'06).* IEEE, 130–132.

Chopra, D., Schulzrinne, H., Marocco, E., and Ivov, E. 2009. Peer-to-peer overlays for real-time communication: Security issues and solutions. *IEEE Commun. Surv. Tut. 11,* 1, 4–12.

Cisco NetFlow. 2010. `http://www.cisco.com/web/go/netflow`. (Last accessed 3/10).

Claffy, K. C., Braun, H.-W., and Polyzos, G. C. 1995. A parameterizable methodology for Internet traffic flow profiling. *IEEE J. Sel. Areas Commun. 13,* 8, 1481–1494.

Claffy, K. C. and McCreary, S. 1999. Internet measurement and data analysis: Passive and active measurement. *Am. Stat. Assoc.*

Constantinou, F. and Mavrommatis, P. 2006. Identifying known and unknown peer-to-peer traffic. In *Proceedings of 5th IEEE International Symposium on Network Computing and Applications (NCA'06)*. IEEE, 93–102.

Couto, A., Nogueira, A., Salvador, P., and Valadas, R. 2008. Identification of peer-to-peer applications' flow patterns. In *Proceedings of the Conference on Next Generation Internet Networks (NGI'08)*. IEEE, 292–299.

Crotti, M., Dusi, M., Gringoli, F., and Salgarelli, L. 2007. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Comput. Commun. Rev. 37,* 1, 5–16.

Crotti, M., Gringoli, F., Pelosato, P., and Salgarelli, L. 2006. A statistical approach to IP-level classification of network traffic. In *Proceedings of the IEEE International Conference on Communications (ICC'06)*, Vol. 1. IEEE, 170–176.

Crovella, M. and Krishnamurthy, B. 2006. *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley & Sons, Inc., New York, NY.

Dainotti, A., de Donato, W., Pescapè, A., and Rossi, P. S. 2008. Classification of network traffic via packet-level hidden markov models. In *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom'08)*. IEEE, 1–5.

Dainotti, A., de Donato, W., Pescapé, A., and Ventre, G. 2009. TIE: A community-oriented traffic classification platform. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA'09)*. Lecture Notes in Computer Science, vol. 5537, Springer-Verlag, Berlin Heidelberg, 64–74.

Dedinski, I., Meer, H. D., Han, L., Mathy, L., Pezaros, D. P., Sventek, J. S., and Xiaoying, Z. 2005. Cross-layer peer-to-peer traffic identification and optimization based on active networking. In *Proceedings of the 7th Annual International Working Conference on Active and Programmable Networks (IWAN'05)*. Springer-Verlag, Berlin Heidelberg, 13–27.

Dewes, C., Wichmann, A., and Feldmann, A. 2003. An analysis of Internet chat systems. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'03)*. ACM, New York, NY, 51–64.

Dhamankar, R. and King, R. 2007. Protocol identification via statistical analysis (PISA). *White Paper, Tipping Point*.

Duffield, N., Lund, C., and Thorup, M. 2005. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Trans. Netw. 13,* 5, 933–946.

Duffield, N. G. 2004. Sampling for passive Internet measurement: A review. *Stati. Sci. 19,* 3, 472–498.

Dusi, M., Crotti, M., Gringoli, F., and Salgarelli, L. 2008. Detection of encrypted tunnels across network boundaries. In *Proceedings of the IEEE International Conference on Communications (ICC'08)*. IEEE, 1738–1744.

Dusi, M., Crotti, M., Gringoli, F., and Salgarelli, L. 2009. Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting. *Comput. Netw. 53,* 1, 81–97.

Early, J. P., Brodley, C. E., and Rosenberg, C. 2003. Behavioral authentication of server flows. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03)*. IEEE Computer Society, Los Alamitos, CA, 46–55.

Ehlert, S. and Petgang, S. 2006. Analysis and signature of Skype VoIP session traffic. Tech. rep. NGNI-SKYPE-06b, Fraunhofer FOKUS, Berlin, Germany. July.

Endace. 2011. Enterprise network monitoring tools–network security system–application performance monitoring. `http://www.endace.com`. (Last accessed 7/11).

Erman, J., Arlitt, M., and Mahanti, A. 2006a. Traffic classification using clustering algorithms. In *Proceedings of the ACM SIGCOMM Workshop on Mining Network Data (MineNet'06)*. ACM, New York, NY, 281–286.

Erman, J., Mahanti, A., and Arlitt, M. 2006b. Internet traffic identification using machine learning. In *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom'06)*. IEEE, 1–6.

Erman, J., Mahanti, A., Arlitt, M., Cohen, I., and Williamson, C. 2007a. Offline/realtime traffic classification using semi-supervised learning. *Perform. Eval. 64,* 9-12, 1194–1213.

Erman, J., Mahanti, A., Arlitt, M., and Williamson, C. 2007b. Identifying and discriminating between web and peer-to-peer traffic in the network core. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM Press, New York, NY, 883–892.

Este, A., Gargiulo, F., Gringoli, F., Salgarelli, L., and Sansone, C. 2008. Pattern recognition approaches for classifying IP flows. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR & SPR'08)*. Lecture Notes in Computer Science, vol. 5342, Springer-Verlag, Berlin Heidelberg, 885–895.

Este, A., Gringoli, F., and Salgarelli, L. 2009. Support vector machines for TCP traffic classification. *Comput. Netw. 53,* 14, 2476–2490.

Ettercap. 2010. `http://ettercap.sourceforge.net`. (Last accessed 3/10).

Finamore, A., Mellia, M., Meo, M., and Rossi, D. 2009. KISS: Stochastic packet inspection. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA'09)*. Lecture Notes in Computer Science, vol. 5537, Springer-Verlag, Berlin Heidelberg, 117–125.

Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., and Diot, C. 2003. Packet-level traffic measurements from the Sprint IP backbone. *IEEE Netw. 17,* 6, 6–16.

Freire, E. P., Ziviani, A., and Salles, R. M. 2008a. Detecting Skype flows in web traffic. In *Proceedings of the IEEE Network Operations and Management Symposium (NOMS'08)*. IEEE, 89–96.

Freire, E. P., Ziviani, A., and Salles, R. M. 2008b. Detecting VoIP calls hidden in web traffic. *IEEE Trans. Netw. Service Manag. 5,* 4, 204–214.

Freire, M. M., Carvalho, D. A., and Pereira, M. 2009. Detection of encrypted traffic in eDonkey network through application signatures. In *Proceedings of the 1st International Conference on Advances in P2P Systems (AP2PS'09)*. IEEE Computer Society Press, Los Alamitos, CA, 174–179.

Gerber, A., Houle, J., Nguyen, H., Roughan, M., and Sen, S. 2003. P2P, the gorilla in the cable. In *Proceedings of the National Cable & Telecommunications Association (NCTA)*. 8–11.

Gomes, J. V. P., Inácio, P. R. M., Freire, M. M., Pereira, M., and Monteiro, P. P. 2008. Analysis of peer-to-peer traffic using a behavioural method based on entropy. In *Proceedings of the 27th IEEE International Performance Computing and Communications Conference (IPCCC'08)*. IEEE Computer Society Press, Los Alamitos, CA, 201–208.

Gonzá1ez-Castaño, F. J., Rodríguez-Hernández, P. S., Martínez-Álvarez, R. P., Gómez, A., López-Cabido, I., and Villasuso-Barreiro, J. 2006. Support vector machine detection of peer-to-peer traffic. In *Proceedings of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA'06)*. IEEE, 103–108.

Gringoli, F., Salgarelli, L., Dusi, M., Cascarano, N., Risso, F., and Claffy, K. C. 2009. GT: Picking up the truth from the ground for Internet traffic. *ACM SIGCOMM Comput. Commun. Rev. 39,* 5, 13–18.

Guo, Z. and Qiu, Z. 2008. Identification peer-to-peer traffic for high speed networks using packet sampling and application signatures. In *Proceedings of the 9th International Conference on Signal Processing (ICSP'08)*. IEEE, 2013–2019.

Haffner, P., Sen, S., Spatscheck, O., and Wang, D. 2005. ACAS: Automated construction of application signatures. In *Proceedings of the ACM SIGCOMM Workshop on Mining Network Data (MineNet'05)*. ACM, New York, NY, 197–202.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl. 11,* 1, 10–18.

HU, Y., CHIU, D.-M., AND LUI, J. C. S. 2008. Application identification based on network behavioral profiles. In *Proceedings of the 16th International Workshop on Quality of Service (IWQoS'08)*. IEEE, 219–228.

HU, Y., CHIU, D.-M., AND LUI, J. C. S. 2009. Profiling and identification of P2P traffic. *Comput. Netw. 53,* 6, 849–863.

HUANG, N.-F., JAI, G.-Y., AND CHAO, H.-C. 2008. Early identifying application traffic with application characteristics. In *Proceedings of the IEEE International Conference on Communications (ICC'08)*. IEEE, 5788–5792.

IANA. 2011. Port numbers. `http://www.iana.org`. (Last accessed 6/11).

IETF. 2008. Specification of the IP flow information export (IPFIX) protocol for the exchange of IP traffic flow information. *RFC 5101*. `http://tools.ietf.org/html/rfc5101`.

ILIOFOTOU, M., KIM, H.-C., FALOUTSOS, M., MITZENMACHER, M., PAPPU, P., AND VARGHESE, G. 2009. Graph-based P2P traffic classification at the Internet backbone. In *Proceedings of the 28th IEEE International Conference on Computer Communications Workshops (InfoCom'09)*. IEEE Press, Piscataway, NJ, 37–42.

ILIOFOTOU, M., PAPPU, P., FALOUTSOS, M., MITZENMACHER, M., SINGH, S., AND VARGHESE, G. 2007. Network monitoring using traffic dispersion graphs (TDGs). In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'07)*. ACM, New York, NY, 315–320.

ILIOFOTOU, M., PAPPU, P., FALOUTSOS, M., MITZENMACHER, M., VARGHESE, G., AND KIM, H. 2008. Graption: Automated detection of P2P applications using traffic dispersion graphs (TDGs). Tech. rep. UCR-CS-2008-06080. June.

INOUE, H., JANSENS, D., HIJAZI, A., AND SOMAYAJI, A. 2007. NetADHICT: A tool for understanding network traffic. In *Proceedings of the 21st Large Installation System Administration Conference (LISA'07)*. USENIX Association, 39–47.

IPOQUE. 2011. Bandwidth management with deep packet inspection. `http://www.ipoque.com`. (Last accessed 7/11).

JAIN, R. AND ROUTHIER, S. A. 1986. Packet trains–measurements and a new model for computer network traffic. *IEEE J. Sel. Areas Commun. 4,* 6, 986–995.

JOHN, W. AND TAFVELIN, S. 2008. Heuristics to classify Internet backbone traffic based on connection patterns. In *Proceedings of the International Conference on Information Networking (ICOIN'08)*. IEEE, 1–5.

JOHNSON, M. E., MCGUIRE, D., AND WILLEY, N. D. 2008. The evolution of the peer-to-peer file sharing industry and the security risks for users. In *Proceedings of the Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS'08)*. IEEE Computer Society, Washington, DC.

JOHNSON, M. E., MCGUIRE, D., AND WILLEY, N. D. 2009. Why file sharing networks are dangerous? *Commun. ACM 52,* 2, 134–138.

JURGA, R. E. AND HULBÓJ, M. M. 2007. Packet sampling for network monitoring. Tech. rep., CERN — HP Procurve openlab project. Dec.

KARAGIANNIS, T., BROIDO, A., BROWNLEE, N., CLAFFY, K., AND FALOUTSOS, M. 2004a. File-sharing in the Internet: A characterization of P2P traffic in the backbone. Tech. rep.

KARAGIANNIS, T., BROIDO, A., BROWNLEE, N., CLAFFY, K. C., AND FALOUTSOS, M. 2004b. Is P2P dying or just hiding? In *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom'04)*, Vol. 3. IEEE Computer Society Press, Piscataway, NJ, 1532–1538.

KARAGIANNIS, T., FALOUTSOS, A. B. M., AND CLAFFY, K. C. 2004c. Transport layer identification of P2P traffic. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'04)*. ACM, New York, NY, 121–134.

KARAGIANNIS, T., PAPAGIANNAKI, K., AND FALOUTSOS, M. 2005a. BLINC: Multilevel traffic classification in the dark. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Vol. 35. ACM, New York, NY, 229–240.

KARAGIANNIS, T., RODRIGUEZ, P., AND PAPAGIANNAKI, K. 2005b. Should Internet service providers fear peer-assisted content distribution? In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'05)*. USENIX Association, Berkeley, CA, 63–76.

KIM, H., CLAFFY, K. C., FOMENKOV, M., BARMAN, D., FALOUTSOS, M., AND LEE, K. 2008. Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proceedings of the ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT'08)*. ACM, New York, NY, 1–12.

KIM, H.-C., FOMENKOV, M., BROWNLEE, N., CLAFFY, K. C., BARMAN, D., AND FALOUTSOS, M. 2007. Comparison of Internet traffic classification tools. In *Proceedings of the Workshop on Application Classification and Identification (WACI)*.

KIND, A., DIMITROPOULOS, X., DENAZIS, S., AND CLAISE, B. 2008. Advanced network monitoring brings life to the awareness plane. *IEEE Commun. Mag. 46,* 10, 140–146.

KRISHNAMURTHY, B. AND WANG, J. 2002. Traffic classification for application specific peering. In *Proceedings of the 2nd ACM SIGCOMM Internet Measurement Workshop (IMW'02)*. ACM, New York, NY, 179–180.

KUMAR, S., DHARMAPURIKAR, S., YU, F., CROWLEY, P., AND TURNER, J. 2006. Algorithms to accelerate multiple regular expressions matching for deep packet inspection. *ACM SIGCOMM Comput. Commun. Rev. 36,* 4, 339–350.

L7-FILTER. 2010. L7-filter, application layer packet classifier for Linux. `http://l7-filter.sourceforge.net`. (Last accessed 3/10).

L7-NETPDLCLASSIFIER. 2010. Tools for L2-L7 traffic classification. `http://netgroup.polito.it/research-projects/l7-traffic-classification/`. (Last accessed 3/10).

LAKHINA, A., CROVELLA, M., AND DIOT, C. 2005. Mining anomalies using traffic feature distributions. *ACM SIGCOMM Comput. Commun. Rev. 35,* 4, 217–228.

LAWTON, G. 2004. Is peer-to-peer secure enough for corporate use? *IEEE Comput. 37,* 1, 22–25.

LEIBOWITZ, N., BERGMAN, A., BEN-SHAUL, R., AND SHAVIT, A. 2002. Are file swapping networks cacheable? Characterizing P2P traffic. In *Proceedings of the 7th International Workshop on Web Content Caching and Distribution (WCW)*.

LI, T., GUAN, Z., AND WU, X. 2007. Modeling and analyzing the spread of active worms based on P2P systems. *Comput. Security 26,* 3, 213–218.

LI, W., CANINI, M., MOORE, A. W., AND BOLLA, R. 2009. Efficient application identification and the temporal and spatial stability of classification schema. *Comput. Netw. 53,* 6, 790–809.

LIN, Y.-D., LU, C.-N., LAI, Y.-C., PENG, W.-H., AND LIN, P.-C. 2009. Application classification using packet size distribution and port association. *J. Netw. Comput. Appl. 32,* 5, 1023–1030.

LIU, H., FENG, W., HUANG, Y., AND LI, X. 2007. A peer-to-peer traffic identification method using machine learning. In *Proceedings of the International Conference on Networking, Architecture, and Storage (NAS'07)*. IEEE, 155–160.

MA, J., LEVCHENKO, K., KREIBICH, C., SAVAGE, S., AND VOELKER, G. M. 2006. Unexpected means of protocol inference. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'06)*. ACM, New York, NY, 313–326.

MADHUKAR, A. AND WILLIAMSON, C. 2006. A longitudinal study of P2P traffic classification. In *Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'06)*. IEEE Computer Society, Washington, DC, 179–188.

MAKHOUL, J., KUBALA, F., SCHWARTZ, R., AND WEISCHEDEL, R. 1999. Performance measures for information extraction. In *Proceedings of the DARPA Broadcast News Workshop*. 249–252.

MANTIA, G. L., ROSSI, D., FINAMORE, A., MELLIA, M., AND MEO, M. 2010. Stochastic packet inspection for TCP traffic. In *Proceedings of the IEEE International Conference on Communications (ICC'10)*. IEEE, 1–6.

MAPI. 2010. MAPI, monitoring API. `http://mapi.uninett.no`. (Last accessed 3/10).

MCGREGOR, A., HALL, M., LORIER, P., AND BRUNSKILL, J. 2004. Flow clustering using machine learning techniques. In *Proceedings of the Passive and Active Measurement Workshop (PAM'04)*. Lecture Notes in Computer Science, vol. 3015, Springer-Verlag, Berlin Heidelberg, 205–214.

MCGREGOR, T. 2002. Quality in measurement: Beyond the deployment barrier. In *Proceedings of the Symposium on Applications and the Internet Workshops (SAINT-W'02)*. IEEE Computer Society, Washington, DC, 66–73.

MOORE, A. W. AND PAPAGIANNAKI, K. 2005. Toward the accurate identification of network applications. In *Proceedings of the Passive and Active Measurement Conference (PAM'05)*. Lecture Notes in Computer Science, vol. 3431. Springer-Verlag, Berlin Heidelberg, 41–54.

MOORE, A. W. AND ZUEV, D. 2005. Internet traffic classification using bayesian analysis techniques. *ACM SIGMETRICS Perform. Eval. Rev. 33,* 1, 50–60.

MOORE, A. W., ZUEV, D., AND CROGAN, M. L. 2005. Discriminators for use in flow-based classification. Tech. rep. RR-05-13, Intel Research, Cambridge, U.K. Aug.

MOORE, D., KEYS, K., KOGA, R., LAGACHE, E., AND CLAFFY, K. C. 2001. The CoralReef software suite as a tool for system and network administrators. In *Proceedings of the 15th USENIX System Administration Conference (LISA'01)*. USENIX Association, Berkeley, CA, 133–144.

MU, J., SEZER, S., DOUGLAS, G., BURNS, D., GARCIA, E., HUTTON, M., AND CACKOVIC, K. 2007. Accelerating pattern matching for DPI. In *Proceedings of the IEEE International Symposium on System-on-Chip (SOC'07)*. IEEE, 83–86.

MURRAY, M. AND CLAFFY, K. C. 2001. Measuring the immeasurable: Global Internet measurement infrastructure. In *Proceedings of the Passive and Active Measurement Workshop (PAM'01)*. 159–167.

NAPATECH. 2011. Intelligent real-time network analysis. http://www.napatech.com. (Last accessed 7/11).

NETADHICT. 2010. http://www.ccsl.carleton.ca/software/netadhict/. (Last accessed 3/10).

NETBEE. 2010. The NetBee library. http://www.nbee.org. (Last accessed 3/10).

NETPDL. 2010. http://www.nbee.org/netpdl. (Last accessed 3/10).

NGUYEN, T. T. T. AND ARMITAGE, G. 2006. Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world IP networks. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN'06)*. IEEE, 369–376.

NGUYEN, T. T. T. AND ARMITAGE, G. 2008a. Clustering to assist supervised machine learning for real-time IP traffic classification. In *Proceedings of the IEEE International Conference on Communications (ICC'08)*. IEEE, 5857–5862.

NGUYEN, T. T. T. AND ARMITAGE, G. 2008b. A survey of techniques for Internet traffic classification using machine learning. *IEEE Commun. Surveys Tuts. 10,* 4, 56–76.

NLANR. 2010. NLANR/MNA home page. http://www.nlanr.net. (Last accessed 3/10).

NOGUEIRA, A., SALVADOR, P., COUTO, A., AND VALADAS, R. 2009. Towards the on-line identification of peer-to-peer flow patterns. *J. Netw. 4,* 2, 108–118.

NOGUEIRA, A., SALVADOR, P., AND VALADAS, R. 2007. A framework for detecting internet applications. In *Proceedings of the International Conference on Information Networking (ICOIN'07)*. Springer-Verlag, Berlin Heidelberg, 455–464.

OHM, P., SICKER, D. C., AND GRUNWALD, D. 2007. Legal issues surrounding monitoring during network research. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'07)*. ACM, New York, NY, 141–148.

OHZAHATA, S., HAGIWARA, Y., TERADA, M., AND KAWASHIMA, K. 2005. A traffic identification method and evaluations for a pure P2P application. In *Proceedings of the Passive and Active Measurement Conference (PAM'05)*. Lecture Notes in Computer Science, vol. 3431, Springer-Verlag, Berlin Heidelberg, 55–68.

OLSON, D. L. AND DELEN, D. 2008. *Advanced Data Mining Techniques* 1st Ed. Springer.

OPENDPI. 2010. OpenDPI - the open source deep packet inspection engine. http://www.opendpi.org. (Last accessed 3/10).

PALMIERI, F. AND FIORE, U. 2009. A nonlinear, recurrence-based approach to traffic classification. *Comput. Netw. 53,* 6, 761–773.

PARK, B.-C., WON, Y. J., KIM, M.-S., AND HONG, J. W. 2008. Towards automated application signature generation for traffic identification. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'08)*. IEEE, 160–167.

PAXSON, V. 2004. Strategies for sound Internet measurement. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'04)*. ACM, New York, NY, 263–271.

PERÉNYI, M., DANG, T. D., GEFFERTH, A., AND MOLNÁR, S. 2006. Identification and analysis of peer-to-peer traffic. *J. Commun. 1,* 7, 36–46.

PLONKA, D. 2000. FlowScan: A network traffic flow reporting and visualization tool. In *Proceedings of the 14th USENIX System Administration Conference (LISA'00)*. USENIX Association, Berkeley, CA, 305–317.

RAAHEMI, B., KOUZNETSOV, A., HAYAJNEH, A., AND RABINOVITCH, P. 2008a. Classification of peer-to-peer traffic using incremental neural networks (fuzzy ARTMAP). In *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE'08)*. IEEE, 719–724.

RAAHEMI, B., ZHONG, W., AND LIU, J. 2008b. Peer-to-peer traffic identification by mining IP layer data streams using concept-adapting very fast decision tree. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'08)*, Vol. 1. IEEE, 525–532.

RANJAN, S., SHAH, S., NUCCI, A., MUNAFÒ, M., CRUZ, R., AND MUTHUKRISHNAN, S. 2007. DoWitcher: Effective worm detection and containment in the internet core. In *Proceedings of the 26th IEEE International Conference on Computer Communications (InfoCom'07)*. IEEE, 2541–2545.

RISSO, F. AND BALDI, M. 2006. NetPDL: An extensible XML-based language for packet header description. *Comput. Netw. 50,* 5, 688–706.

RISSO, F., BALDI, M., MORANDI, O., BALDINI, A., AND MONCLUS, P. 2008. Lightweight, payload-based traffic classification: An experimental evaluation. In *Proceedings of the IEEE International Conference on Communications (ICC'08)*. IEEE, 5869–5875.

ROMIG, S., FULLMER, M., AND LUMAN, R. 2000. The OSU flow-tools package and CISCO NetFlow logs. In *Proceedings of the 14th USENIX System Administration Conference (LISA'00)*. USENIX Association, Berkeley, CA, 291–303.

SALGARELLI, L., GRINGOLI, F., AND KARAGIANNIS, T. 2007. Comparing traffic classifiers. *ACM SIGCOMM Comput. Commun. Rev. 37,* 3, 65–68.

Saroiu, S., Gummadi, K. P., Dunn, R. J., Gribble, S. D., and Levy, H. M. 2002a. An analysis of Internet content delivery systems. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02)*, Vol. 36. ACM, New York, NY, 315–327.

Saroiu, S., Gummadi, P. K., and Gribble, S. D. 2002b. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the Multimedia Computing and Networking (MMCN'02)*. ACM, New York, NY.

Saroiu, S., Gummadi, P. K., and Gribble, S. D. 2003. Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Syst. J. 9,* 2, 170–184.

Schmidt, Ş. E. G. and Soysal, M. 2006. An intrusion detection based approach for the scalable detection of P2P traffic in the national academic backbone network. In *Proceedings of the International Symposium on Computer Networks (ISCN'06)*. IEEE, 128–133.

Schulze, H. and Mochalski, K. 2007. Internet study 2007. Tech. rep., ipoque.

Schulze, H. and Mochalski, K. 2009. Internet study 2008/2009. Tech. rep., ipoque.

Seedorf, J. 2006. Security challenges for peer-to-peer SIP. *IEEE Netw. 20,* 5, 38–45.

Sen, S., Spatscheck, O., and Wang, D. 2004. Accurate, scalable in-network identification of P2P traffic using application signatures. In *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*. ACM, New York, NY, 512–521.

Sen, S. and Wang, J. 2004. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Trans. Netw. 12,* 2, 219–232.

Sena, G. G. and Belzarena, P. 2009. Early traffic classification using support vector machines. In *Proceedings of the 5th International Latin American Networking Conference (LANC'09)*. ACM, New York, NY, 60–66.

Singh, S., Estan, C., Varghese, G., and Savage, S. 2004. Automated worm fingerprinting. In *Proceedings of the 6th Symposium on Operating Systems Design & Implementation (OSDI'04)*. USENIX Association, Berkeley, CA, USA, 45–60.

Smith, F. D., Campos, F. H., Jeffay, K., and Ott, D. 2001. What TCP/IP protocol headers can tell us about the Web. *ACM SIGMETRICS Perform. Eval. Rev. 29,* 1, 245–256.

Smith, R., Estan, C., Jha, S., and Kong, S. 2008. Deflating the big bang: Fast and scalable deep packet inspection with extended finite automata. *ACM SIGCOMM Comput. Commun. Rev. 38,* 4, 207–218.

Snort. 2010. `http://www.snort.org`. (Last accessed 3/10).

Soewito, B., Mahajan, A., Weng, N., and Wang, H. 2009. High-speed string matching for network intrusion detection. *Int. J. Commun. Netw. Distrib. Syst. 3,* 4, 319–339.

Soysal, M. and Schmidt, E. G. 2007. An accurate evaluation of machine learning algorithms for flow-based P2P traffic detection. In *Proceedings of the 22nd International Symposium on Computer and Information Sciences (ISCIS'07)*. IEEE.

Sperotto, A., Sadre, R., van Vliet, F., and Pras, A. 2009. A labeled data set for flow-based intrusion detection. In *Proceedings of the 9th IEEE International Workshop on IP Operations and Management (IPOM'09)*. Lecture Notes in Computer Science, vol. 5843, Springer-Verlag, Berlin Heidelberg, 39–50.

Spognardi, A., Lucarelli, A., and Pietro, R. D. 2005. A methodology for P2P file-sharing traffic detection. In *Proceedings of the 2nd International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'05)*. IEEE Computer Society, Washington, DC, 52–61.

Stefanowski, J. and Wilk, S. 2009. Extending rule-based classifiers to improve recognition of imbalanced classes. In *Advances in Data Management*, Z. W. Ras and A. Dardzinska, Eds., Studies in Computational Intelligence, vol. 223, Springer-Verlag, Berlin Heidelberg, 131–154.

Strayer, T., Armitage, G., Allman, M., Moore, A. W., Jin, S., and Bellovin, S. 2008. IMRG workshop on application classification and identification report. *ACM SIGCOMM Comput. Commun. Rev. 38,* 3, 87–90.

Szabó, G., Orincsay, D., Malomsoky, S., and Szabó, I. 2008. On the validation of traffic classification algorithms. In *Proceedings of the Passive and Active Measurement Conference (PAM'08)*. Lecture Notes in Computer Science, vol. 4979, Springer-Verlag, Berlin Heidelberg, 72–81.

Szabó, G., Szabó, I., and Orincsay, D. 2007. Accurate traffic classification. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*. IEEE, 1–8.

tcpdump. 2011. TCPDUMP/LIBPCAP public repository. `http://www.tcpdump.org`. (Last accessed 7/10).

TIE. 2010. TIE, traffic identification engine. `http://tie.comics.unina.it`. (Last accessed 7/10).

Turkett, W. H., Karode, A. V., and Fulp, E. W. 2008. In-the-dark network traffic classification using support vector machines. In *Proceedings of the 20th National Conference on Innovative Applications of Artificial Intelligence (IAAI'08)*. AAAI Press, 1745–1750.

VALENTI, S., ROSSI, D., MEO, M., MELLIA, M., AND BERMOLEN, P. 2009. Accurate, fine-grained classification of P2P-TV applications by simply counting packets. In *Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA'09)*, Lecture Notes in Computer Science, vol. 5537, Springer-Verlag, Berlin, Heidelberg, 84–92.

WANG, Y. 2008. *Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection*. Premier Reference Source. Information Science Reference.

WANG, Y.-H., GAU, V., BOSAW, T., HWANG, J.-N., LIPPMAN, A., LIEBENNAN, D., AND WU, I.-C. 2008. Generalization performance analysis of flow-based peer-to-peer traffic identification. In *Proceedings of the IEEE Workshop on Machine Learning for Signal Processing (MLSP'08)*. IEEE, 267–272.

WEISS, G. M. 2004. Mining with rarity: A unifying framework. *ACM SIGKDD Explor. Newsl. 6,* 1, 7–19.

WILDPACKETS. 2011. WildPackets: Network analyzer, voip monitoring, protocol analysis. `http://www.wildpackets.com`. (Last accessed 7/11).

WILLIAMS, N., ZANDER, S., AND ARMITAGE, G. 2006. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Comput. Commun. Rev. 36,* 5, 5–16.

WILLIAMSON, C. 2001. Internet traffic measurement. *IEEE Internet Comput. 5,* 6, 70–74.

WINDUMP. 2011. tcpdump for Windows using WinPcap. `http://www.winpcap.org/windump/`. (Last accessed 7/11).

WINPCAP. 2011. The industry-standard windows packet capture library. `http://www.winpcap.org`. (Last accessed 7/11).

WIRESHARK. 2010. Wireshark, go deep. `http://www.wireshark.org`. (Last accessed 3/10).

WRIGHT, C. V., MONROSE, F., AND MASSON, G. M. 2006. On inferring application protocol behaviors in encrypted network traffic. *J. Mach. Learn. Res. 7*, 2745–2769.

XU, K., LIU, J., AND WANG, H. 2008. Tod-cache: Peer-to-peer traffic management and optimization using combined caching and redirection. In *Proceedings of the IEEE Global Telecommunications Conference (GlobeCom'08)*. IEEE, 1–5.

XUSHENG, Z. AND ZHIMING, W. 2009. Application of markov chain in IP traffic classification. In *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC'09)*, Vol. 2. IEEE Computer Society, 688–691.

YEGNESWARAN, V., GIFFIN, J. T., BARFORD, P., AND JHA, S. 2005. An architecture for generating semantics-aware signatures. In *Proceedings of the 14th USENIX Security Symposium (SSYM'05)*. USENIX Association, Berkeley, CA, 97–112.

YU, F. 2006. High speed deep packet inspection with hardware support. Ph.D. dissertation, EECS Department, University of California, Berkeley, CA.

ZANDER, S., NGUYEN, T., AND ARMITAGE, G. 2005a. Automated traffic classification and application identification using machine learning. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN'2005)*. IEEE, 250–257.

ZANDER, S., NGUYEN, T., AND ARMITAGE, G. 2005b. Self-learning IP traffic classification based on statistical flow characteristics. In *Proceedings of the Passive and Active Measurement Conference (PAM'05)*. Lecture Notes in Computer Science, vol. 3431. Springer-Verlag, Berlin Heidelberg, 325–328.

ZHOU, L., ZHANG, L., MCSHERRY, F., IMMORLICA, N., COSTA, M., AND CHIEN, S. 2005. A first look at peer-to-peer worms: Threats and defenses. In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*. Lecture Notes in Computer Science, vol. 3640, Springer, Berlin Heidelberg, 24–35.

ZUEV, D. AND MOORE, A. W. 2005. Traffic classification using a statistical approach. In *Proceedings of the Passive and Active Measurement Conference (PAM'05)*. Lecture Notes in Computer Science, vol. 3431, Springer-Verlag, Berlin Heidelberg, 321–324.

Received May 2010; revised August, December 2011; accepted January 2012